TECHNISCHE UNIVERSITÄT DRESDEN

Computational Logic ∴ Group

Hannes Strass

Faculty of Computer Science, Institute of Artificial Intelligence, Computational Logic Group

Approximation Fixpoint Theory

Lecture 12, 23rd Jan 2023 // Foundations of Knowledge Representation, WS 2022/23

Motivation: Objective

Goal: Define semantics for (rule-based) KR formalisms in the presence of:

Recursion

- transitive closure
- indirect effects of actions

Motivation: Objective

Goal: Define semantics for (rule-based) KR formalisms in the presence of:

Recursion

- transitive closure
- indirect effects of actions

Negation

- shorter and more intuitive descriptions
- defaults and assumptions (e.g. closed world, non-effects of actions)

Motivation: Objective

Goal: Define semantics for (rule-based) KR formalisms in the presence of:

Recursion

- transitive closure
- indirect effects of actions

Negation

- shorter and more intuitive descriptions
- defaults and assumptions (e.g. closed world, non-effects of actions)

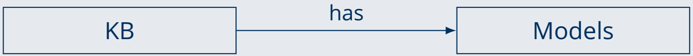
Recursion Through Negation

- mutually exclusive alternatives
- non-deterministic effects of actions

Motivation: Basic Idea

Approximation Fixpoint Theory

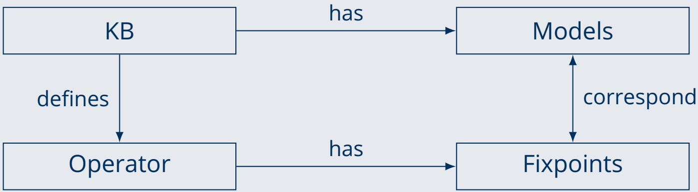
- Framework for studying semantics of (non-monotonic) KR formalisms
- Due to Denecker, Marek, and Truszczyński [2000, 2003, 2004]
- Based on lattice theory and fixpoint theory:



Motivation: Basic Idea

Approximation Fixpoint Theory

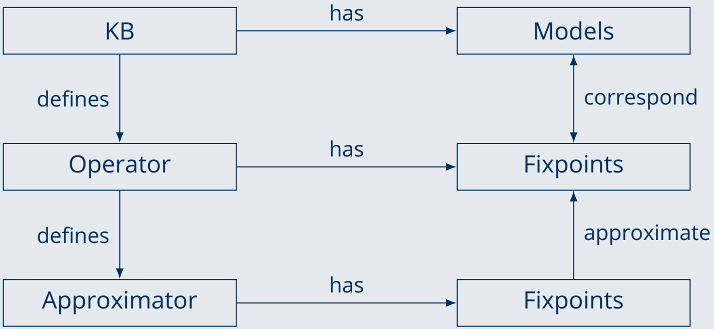
- Framework for studying semantics of (non-monotonic) KR formalisms
- Due to Denecker, Marek, and Truszczyński [2000, 2003, 2004]
- Based on lattice theory and fixpoint theory:



Motivation: Basic Idea

Approximation Fixpoint Theory

- Framework for studying semantics of (non-monotonic) KR formalisms
- Due to Denecker, Marek, and Truszczyński [2000, 2003, 2004]
- Based on lattice theory and fixpoint theory:



Motivation: History and Context

Approximation Fixpoint Theory

… emerged from similarities in the semantics of

- Default Logic [Reiter, 1980]
- Autoepistemic Logic [Moore, 1985]
- Logic Programs, in particular Stable Models [Gelfond and Lifschitz, 1988]

… and has since been applied to define/reconstruct semantics of …

- Abstract Argumentation Frameworks
- Abstract Dialectical Frameworks
- Active Integrity Constraints
- Recursive SHACL

Agenda

Preliminaries

Partially Ordered Sets

Definition

A **partially ordered set** is a pair $(L, \leqslant)$ with

- L a set, and (carrier set)
- $\leqslant \subseteq L \times L$ a partial order. (reflexive, antisymmetric, transitive)

Partially Ordered Sets

Definition

A **partially ordered set** is a pair $(L, \leqslant)$ with

- L a set, and (carrier set)
- $\leqslant \subseteq L \times L$ a partial order. (reflexive, antisymmetric, transitive)

A partially ordered set $(L, \leqslant)$ has a

- **bottom element** $\bot \in L$ iff $\bot \leqslant x$ for all $x \in L$,
- **top element** $\top \in L$ iff $x \leqslant \top$ for all $x \in L$.

Partially Ordered Sets

Definition

A **partially ordered set** is a pair $(L, \leqslant)$ with

- L a set, and (carrier set)
- $\leqslant \subseteq L \times L$ a partial order. (reflexive, antisymmetric, transitive)

A partially ordered set $(L, \leqslant)$ has a

- **bottom element** $\bot \in L$ iff $\bot \leqslant x$ for all $x \in L$,
- **top element** $\top \in L$ iff $x \leqslant \top$ for all $x \in L$.

Examples

- $(\mathbb{N}, \leq)$: natural numbers with "usual" ordering, $\bot = 0$, no $\top$
- $(2^S, \subseteq)$: any powerset with subset relation, $\bot = \emptyset$, $\top = S$
- $(\mathbb{N}, |)$: natural numbers with divisibility relation, $\bot = 1$, $\top = 0$

Minimal, Maximal, Least, Greatest

Definition

Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in S$. We say that:

- x is a **minimal element** of S iff for each $y \in S$, $y \leqslant x$ implies $y = x$, dually,
- x is a **maximal element** of S iff for each $y \in S$, $x \leqslant y$ implies $y = x$;

Minimal, Maximal, Least, Greatest

Definition

Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in S$. We say that:

- x is a **minimal element** of S iff for each $y \in S$, $y \leqslant x$ implies $y = x$, dually,
- x is a **maximal element** of S iff for each $y \in S$, $x \leqslant y$ implies $y = x$;
- x is the **least element** of S iff for each $y \in S$, we have $x \leqslant y$, dually,
- x is the **greatest element** of S iff for each $y \in S$, we have $y \leqslant x$.

Minimal, Maximal, Least, Greatest

Definition

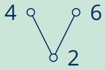
Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in S$. We say that:

- x is a **minimal element** of S iff for each $y \in S$, $y \leqslant x$ implies $y = x$, dually,
- x is a **maximal element** of S iff for each $y \in S$, $x \leqslant y$ implies $y = x$;
- x is the **least element** of S iff for each $y \in S$, we have $x \leqslant y$, dually,
- x is the **greatest element** of S iff for each $y \in S$, we have $y \leqslant x$.

Example

In $(\mathbb{N}, |)$ (natural numbers with divisibility $a \mid b \iff (\exists k \in \mathbb{N}) a \cdot k = b$), …

- the set $\{2, 3, 6\}$ has minimal elements 2 and 3, greatest element 6,
- the set $\{2, 4, 6\}$ has least element 2, and maximal elements 4 and 6.



Least Upper and Greatest Lower Bounds

Definition

Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in L$.

- x is an **upper bound** of S iff for each $s \in S$, we have $s \leqslant x$, dually,
- x is a **lower bound** of S iff for each $s \in S$, we have $x \leqslant s$.

Least Upper and Greatest Lower Bounds

Definition

Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in L$.

- x is an **upper bound** of S iff for each $s \in S$, we have $s \leqslant x$, dually,
- x is a **lower bound** of S iff for each $s \in S$, we have $x \leqslant s$.

The set of all upper bounds of S is denoted by S^u, its lower bounds by S^ℓ.

Least Upper and Greatest Lower Bounds

Definition

Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in L$.

- x is an **upper bound** of S iff for each $s \in S$, we have $s \leqslant x$, dually,
- x is a **lower bound** of S iff for each $s \in S$, we have $x \leqslant s$.

The set of all upper bounds of S is denoted by S^u, its lower bounds by S^ℓ.

- If S^u has a least element $z \in S$, z is the **least upper bound** of S, dually,
- if S^ℓ has a greatest element $z \in S$, z is the **greatest lower bound** of S.

Least Upper and Greatest Lower Bounds

Definition

Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in L$.

- x is an **upper bound** of S iff for each $s \in S$, we have $s \leqslant x$, dually,
- x is a **lower bound** of S iff for each $s \in S$, we have $x \leqslant s$.

The set of all upper bounds of S is denoted by S^u, its lower bounds by S^ℓ.

- If S^u has a least element $z \in S$, z is the **least upper bound** of S, dually,
- if S^ℓ has a greatest element $z \in S$, z is the **greatest lower bound** of S.

We denote the **glb** of $\{x, y\}$ by $x \wedge y$, and the **lub** of $\{x, y\}$ by $x \vee y$.

Least Upper and Greatest Lower Bounds

Definition

Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in L$.

- x is an **upper bound** of S iff for each $s \in S$, we have $s \leqslant x$, dually,
- x is a **lower bound** of S iff for each $s \in S$, we have $x \leqslant s$.

The set of all upper bounds of S is denoted by S^u, its lower bounds by S^ℓ.

- If S^u has a least element $z \in S$, z is the **least upper bound** of S, dually,
- if S^ℓ has a greatest element $z \in S$, z is the **greatest lower bound** of S.

We denote the **glb** of $\{x, y\}$ by $x \wedge y$, and the **lub** of $\{x, y\}$ by $x \vee y$.
We denote the glb of S by $\bigwedge S$, and the lub of S by $\bigvee S$.

Least Upper and Greatest Lower Bounds

Definition

Let $(L, \leqslant)$ be a partially ordered set with $S \subseteq L$ and $x \in L$.

- x is an **upper bound** of S iff for each $s \in S$, we have $s \leqslant x$, dually,
- x is a **lower bound** of S iff for each $s \in S$, we have $x \leqslant s$.

The set of all upper bounds of S is denoted by S^u, its lower bounds by S^ℓ.

- If S^u has a least element $z \in S$, z is the **least upper bound** of S, dually,
- if S^ℓ has a greatest element $z \in S$, z is the **greatest lower bound** of S.

We denote the **glb** of $\{x, y\}$ by $x \wedge y$, and the **lub** of $\{x, y\}$ by $x \vee y$.
We denote the glb of S by $\bigwedge S$, and the lub of S by $\bigvee S$.

Examples

- In $(2^S, \subseteq)$, $\wedge = \cap$ and $\vee = \cup$;
- in $(\mathbb{N}, \mid)$, $\wedge =$ gcd and $\vee =$ lcm, e.g. $4 \vee 6 = 12$ and $23 \wedge 42 = 1$.

(Complete) Lattices

Definition

Let $(L, \leqslant)$ be a partially ordered set.

1. $(L, \leqslant)$ is a **lattice** if and only if for all $x, y \in L$, both $x \wedge y$ and $x \vee y$ exist;

(Complete) Lattices

Definition

Let $(L, \leqslant)$ be a partially ordered set.

1. $(L, \leqslant)$ is a **lattice** if and only if for all $x, y \in L$, both $x \wedge y$ and $x \vee y$ exist;
2. $(L, \leqslant)$ is a **complete lattice** iff for all $S \subseteq L$, both $\bigwedge S$ and $\bigvee S$ exist.

(Complete) Lattices

Definition

Let $(L, \leqslant)$ be a partially ordered set.

1. $(L, \leqslant)$ is a **lattice** if and only if for all $x, y \in L$, both $x \wedge y$ and $x \vee y$ exist;
2. $(L, \leqslant)$ is a **complete lattice** iff for all $S \subseteq L$, both $\bigwedge S$ and $\bigvee S$ exist.

In particular, a complete lattice has $\bigvee \emptyset = \bigwedge L = \bot$ and $\bigwedge \emptyset = \bigvee L = \top$.

(Complete) Lattices

Definition

Let $(L, \leqslant)$ be a partially ordered set.

1. $(L, \leqslant)$ is a **lattice** if and only if for all $x, y \in L$, both $x \wedge y$ and $x \vee y$ exist;
2. $(L, \leqslant)$ is a **complete lattice** iff for all $S \subseteq L$, both $\bigwedge S$ and $\bigvee S$ exist.

In particular, a complete lattice has $\bigvee \emptyset = \bigwedge L = \bot$ and $\bigwedge \emptyset = \bigvee L = \top$.

Examples

- $(2^S, \subseteq)$ is a complete lattice for every set S.
- $(\mathbb{N}, |)$ is a complete lattice.
- $(\{M \subseteq \mathbb{N} \mid M \text{ is finite}\}, \subseteq)$ is a lattice.
- Every lattice $(L, \leqslant)$ with L finite is a complete lattice. (induction on $|S|$)

Further reading: B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Second Edition. Cambridge University Press, 2002

Operators and Their Properties

Definition

Let $(L, \leqslant)$ be a partially ordered set.
An operator $O : L \to L$ is $\leqslant$-**monotone** if and only if for all $x, y \in L$,

$$x \leqslant y \quad \text{implies} \quad O(x) \leqslant O(y)$$

Intuition: Operator application preserves ordering.

Operators and Their Properties

Definition

Let $(L, \leqslant)$ be a partially ordered set.
An operator $O : L \to L$ is $\leqslant$-**monotone** if and only if for all $x, y \in L$,

$$x \leqslant y \quad \text{implies} \quad O(x) \leqslant O(y)$$

Intuition: Operator application preserves ordering.

Example

Consider $(2^{\mathbb{N}}, \subseteq)$ with operator $O : 2^{\mathbb{N}} \to 2^{\mathbb{N}}, \quad M \mapsto \{\prod K \mid K \subseteq M,\ K \text{ finite}\}$.

Operators and Their Properties

Definition

Let $(L, \leqslant)$ be a partially ordered set.
An operator $O : L \to L$ is $\leqslant$-**monotone** if and only if for all $x, y \in L$,

$$x \leqslant y \quad \text{implies} \quad O(x) \leqslant O(y)$$

Intuition: Operator application preserves ordering.

Example

Consider $(2^{\mathbb{N}}, \subseteq)$ with operator $O : 2^{\mathbb{N}} \to 2^{\mathbb{N}}, \quad M \mapsto \{\prod K \mid K \subseteq M,\ K \text{ finite}\}$.

- $O(\{2, 3\}) = \{1, 2, 3, 6\}$ and $O(\{2, 3, 5\}) = \{1, 2, 3, 5, 6, 10, 15, 30\}$.

Operators and Their Properties

Definition

Let $(L, \leqslant)$ be a partially ordered set.
An operator $O : L \to L$ is $\leqslant$-**monotone** if and only if for all $x, y \in L$,

$$x \leqslant y \quad \text{implies} \quad O(x) \leqslant O(y)$$

Intuition: Operator application preserves ordering.

Example

Consider $(2^{\mathbb{N}}, \subseteq)$ with operator $O : 2^{\mathbb{N}} \to 2^{\mathbb{N}}, \quad M \mapsto \{\prod K \mid K \subseteq M,\ K \text{ finite}\}$.

- $O(\{2,3\}) = \{1,2,3,6\}$ and $O(\{2,3,5\}) = \{1,2,3,5,6,10,15,30\}$.
- O is $\subseteq$-monotone:

Operators and Their Properties

Definition

Let $(L, \leqslant)$ be a partially ordered set.
An operator $O : L \to L$ is $\leqslant$-**monotone** if and only if for all $x, y \in L$,

$$x \leqslant y \quad \text{implies} \quad O(x) \leqslant O(y)$$

Intuition: Operator application preserves ordering.

Example

Consider $(2^{\mathbb{N}}, \subseteq)$ with operator $O : 2^{\mathbb{N}} \to 2^{\mathbb{N}}, \quad M \mapsto \{\prod K \mid K \subseteq M,\ K \text{ finite}\}$.

- $O(\{2,3\}) = \{1,2,3,6\}$ and $O(\{2,3,5\}) = \{1,2,3,5,6,10,15,30\}$.
- O is $\subseteq$-monotone:
  - Let $M_1 \subseteq M_2 \subseteq \mathbb{N}$ and consider $k \in O(M_1)$.

Operators and Their Properties

Definition

Let $(L, \leqslant)$ be a partially ordered set.
An operator $O : L \to L$ is $\leqslant$-**monotone** if and only if for all $x, y \in L$,

$$x \leqslant y \quad \text{implies} \quad O(x) \leqslant O(y)$$

Intuition: Operator application preserves ordering.

Example

Consider $(2^{\mathbb{N}}, \subseteq)$ with operator $O : 2^{\mathbb{N}} \to 2^{\mathbb{N}}, \quad M \mapsto \{\prod K \mid K \subseteq M, \ K \text{ finite}\}$.

- $O(\{2,3\}) = \{1,2,3,6\}$ and $O(\{2,3,5\}) = \{1,2,3,5,6,10,15,30\}$.
- O is $\subseteq$-monotone:
  - Let $M_1 \subseteq M_2 \subseteq \mathbb{N}$ and consider $k \in O(M_1)$.
  - Then there is a $K \subseteq M_1$ with $k = \prod K$.

Operators and Their Properties

Definition

Let $(L, \leqslant)$ be a partially ordered set.
An operator $O : L \to L$ is $\leqslant$-**monotone** if and only if for all $x, y \in L$,

$$x \leqslant y \quad \text{implies} \quad O(x) \leqslant O(y)$$

Intuition: Operator application preserves ordering.

Example

Consider $(2^{\mathbb{N}}, \subseteq)$ with operator $O : 2^{\mathbb{N}} \to 2^{\mathbb{N}}, \quad M \mapsto \{\prod K \mid K \subseteq M,\ K \text{ finite}\}$.

- $O(\{2,3\}) = \{1,2,3,6\}$ and $O(\{2,3,5\}) = \{1,2,3,5,6,10,15,30\}$.
- O is $\subseteq$-monotone:
  - Let $M_1 \subseteq M_2 \subseteq \mathbb{N}$ and consider $k \in O(M_1)$.
  - Then there is a $K \subseteq M_1$ with $k = \prod K$.
  - By $K \subseteq M_1 \subseteq M_2$, we get $k \in O(M_2)$.

Fixpoints of Operators

Definition

Let $(L, \leqslant)$ be a partially ordered set and $O : L \to L$ be an operator.

- $x \in L$ is a **fixpoint** of O iff $O(x) = x$;
- $x \in L$ is a **prefixpoint** of O iff $O(x) \leqslant x$;
- $x \in L$ is a **postfixpoint** of O iff $x \leqslant O(x)$.

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Fixpoints of Operators

Definition

Let $(L, \leqslant)$ be a partially ordered set and $O : L \to L$ be an operator.

- $x \in L$ is a **fixpoint** of O iff $O(x) = x$;
- $x \in L$ is a **prefixpoint** of O iff $O(x) \leqslant x$;
- $x \in L$ is a **postfixpoint** of O iff $x \leqslant O(x)$.

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Order-preserving operators on complete lattices have a fixpoint.

Fixpoints of Operators

Definition

Let $(L, \leqslant)$ be a partially ordered set and $O : L \to L$ be an operator.

- $x \in L$ is a **fixpoint** of O iff $O(x) = x$;
- $x \in L$ is a **prefixpoint** of O iff $O(x) \leqslant x$;
- $x \in L$ is a **postfixpoint** of O iff $x \leqslant O(x)$.

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Order-preserving operators on complete lattices have a fixpoint.

Example (Continued.)

Consider $(2^{\mathbb{N}}, \subseteq)$ with operator $O : 2^{\mathbb{N}} \to 2^{\mathbb{N}}, \quad M \mapsto \{\prod K \mid K \subseteq M,\ K \text{ finite}\}$.
O has least and greatest fixpoints: $O(\{1\}) = \{1\}$ and $O(\mathbb{N}) = \mathbb{N}$.

Fixpoints of Operators (2)

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Proof.

Define $A = \{x \in L \mid O(x) \leqslant x\}$ and $a = \bigwedge A$. $(A \neq \emptyset$ as $\top \in A.)$

Fixpoints of Operators (2)

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Proof.

Define $A = \{x \in L \mid O(x) \leqslant x\}$ and $\alpha = \bigwedge A$. ($A \neq \emptyset$ as $\top \in A$.)

- For every $x \in A$, we have $\alpha \leqslant x$ and by monotonicity $O(\alpha) \leqslant O(x) \leqslant x$.

Fixpoints of Operators (2)

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Proof.

Define $A = \{x \in L \mid O(x) \leqslant x\}$ and $\alpha = \bigwedge A$. $(A \neq \emptyset$ as $\top \in A.)$

- For every $x \in A$, we have $\alpha \leqslant x$ and by monotonicity $O(\alpha) \leqslant O(x) \leqslant x$.
- Thus $O(\alpha)$ is a lower bound of A.

Fixpoints of Operators (2)

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Proof.

Define $A = \{x \in L \mid O(x) \leqslant x\}$ and $a = \bigwedge A$. ($A \neq \emptyset$ as $\top \in A$.)

- For every $x \in A$, we have $a \leqslant x$ and by monotonicity $O(a) \leqslant O(x) \leqslant x$.
- Thus $O(a)$ is a lower bound of A.
- Since a is the greatest lower bound of A, we get $O(a) \leqslant a$, that is, $a \in A$.

Fixpoints of Operators (2)

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Proof.

Define $A = \{x \in L \mid O(x) \leqslant x\}$ and $\alpha = \bigwedge A$. $(A \neq \emptyset$ as $\top \in A.)$

- For every $x \in A$, we have $\alpha \leqslant x$ and by monotonicity $O(\alpha) \leqslant O(x) \leqslant x$.
- Thus $O(\alpha)$ is a lower bound of A.
- Since α is the greatest lower bound of A, we get $O(\alpha) \leqslant \alpha$, that is, $\alpha \in A$.
- Furthermore, monotonicity yields $O(O(\alpha)) \leqslant O(\alpha)$, whence $O(\alpha) \in A$.

Fixpoints of Operators (2)

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Proof.

Define $A = \{x \in L \mid O(x) \leqslant x\}$ and $a = \bigwedge A$. ($A \neq \emptyset$ as $\top \in A$.)

- For every $x \in A$, we have $a \leqslant x$ and by monotonicity $O(a) \leqslant O(x) \leqslant x$.
- Thus $O(a)$ is a lower bound of A.
- Since a is the greatest lower bound of A, we get $O(a) \leqslant a$, that is, $a \in A$.
- Furthermore, monotonicity yields $O(O(a)) \leqslant O(a)$, whence $O(a) \in A$.
- Since a is a lower bound of A, we get $a \leqslant O(a)$, thus $O(a) = a$.

Fixpoints of Operators (2)

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Proof.

Define $A = \{x \in L \mid O(x) \leqslant x\}$ and $\alpha = \bigwedge A$. ($A \neq \emptyset$ as $\top \in A$.)

- For every $x \in A$, we have $\alpha \leqslant x$ and by monotonicity $O(\alpha) \leqslant O(x) \leqslant x$.
- Thus $O(\alpha)$ is a lower bound of A.
- Since α is the greatest lower bound of A, we get $O(\alpha) \leqslant \alpha$, that is, $\alpha \in A$.
- Furthermore, monotonicity yields $O(O(\alpha)) \leqslant O(\alpha)$, whence $O(\alpha) \in A$.
- Since α is a lower bound of A, we get $\alpha \leqslant O(\alpha)$, thus $O(\alpha) = \alpha$.
- Greatest fixpoint β is obtained dually: $B = \{x \in L \mid x \leqslant O(x)\}$, $\beta = \bigvee B$. $\square$

Fixpoints of Operators (2)

Theorem (Knaster/Tarski)

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a monotone operator.
Then the set F of fixpoints of O has a least element and a greatest element.

Proof.

Define $A = \{x \in L \mid O(x) \leqslant x\}$ and $\alpha = \bigwedge A$. $(A \neq \emptyset$ as $\top \in A.)$

- For every $x \in A$, we have $\alpha \leqslant x$ and by monotonicity $O(\alpha) \leqslant O(x) \leqslant x$.
- Thus $O(\alpha)$ is a lower bound of A.
- Since α is the greatest lower bound of A, we get $O(\alpha) \leqslant \alpha$, that is, $\alpha \in A$.
- Furthermore, monotonicity yields $O(O(\alpha)) \leqslant O(\alpha)$, whence $O(\alpha) \in A$.
- Since α is a lower bound of A, we get $\alpha \leqslant O(\alpha)$, thus $O(\alpha) = \alpha$.
- Greatest fixpoint β is obtained dually: $B = \{x \in L \mid x \leqslant O(x)\}$, $\beta = \bigvee B$. $\square$

$(F, \leqslant)$ is a complete lattice: for $G \subseteq F$, take $([\bigvee G, \bigvee L], \leqslant)$ and $([\bigwedge L, \bigwedge G], \leqslant)$.

Fixpoints of Operators (3)

Nice to know there is one, but how do we get there?

Theorem

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be a $\leqslant$-monotone operator. For ordinals α, β, define

$$\begin{aligned} O^0(\bot) &= \bot \\ O^{\alpha+1}(\bot) &= O(O^\alpha(\bot)) && \text{for successor ordinals} \\ O^\beta(\bot) &= \bigvee \left\{ O^\alpha(\bot) \,\middle|\, \alpha < \beta \right\} && \text{for limit ordinals} \end{aligned}$$

Then for some ordinal α, the element $O^\alpha(\bot)$ is a fixpoint of O.

Example (Continued.)

Consider $(2^{\mathbb{N}}, \subseteq)$ with operator $O : 2^{\mathbb{N}} \to 2^{\mathbb{N}}, \quad M \mapsto \{\prod K \mid K \subseteq M,\ K \text{ finite}\}$.
We obtain the chain $O^0(\emptyset) = \emptyset \rightsquigarrow O^1(\emptyset) = \{1\} \rightsquigarrow O^2(\emptyset) = O(\{1\}) = \{1\}$.

Definite Logic Programs

Consider a set $\mathcal{A}$ of propositional atoms.

Definition

A **definite logic program** over $\mathcal{A}$ is a set P of rules of the form

$$a_0 \leftarrow a_1, \ldots, a_m$$

for $a_0, \ldots, a_m \in \mathcal{A}$ with $0 \leq m$.

A set of definite Horn clauses (exactly one positive literal).

Definite Logic Programs

Consider a set $\mathcal{A}$ of propositional atoms.

Definition

A **definite logic program** over $\mathcal{A}$ is a set P of rules of the form

$$a_0 \leftarrow a_1, \ldots, a_m$$

for $a_0, \ldots, a_m \in \mathcal{A}$ with $0 \leq m$.

A set of definite Horn clauses (exactly one positive literal).

Definition

- A set $S \subseteq \mathcal{A}$ is **closed** under a rule $a \leftarrow a_1, \ldots, a_m$ if and only if $\{a_1, \ldots, a_m\} \subseteq S$ implies $a \in S$.

Definite Logic Programs

Consider a set $\mathcal{A}$ of propositional atoms.

Definition

A **definite logic program** over $\mathcal{A}$ is a set P of rules of the form

$$a_0 \leftarrow a_1, \ldots, a_m$$

for $a_0, \ldots, a_m \in \mathcal{A}$ with $0 \leq m$.

A set of definite Horn clauses (exactly one positive literal).

Definition

- A set $S \subseteq \mathcal{A}$ is **closed** under a rule $a \leftarrow a_1, \ldots, a_m$ if and only if $\{a_1, \ldots, a_m\} \subseteq S$ implies $a \in S$.
- The **least model** of P is the $\subseteq$-least set that is closed under all rules in P.

Definite Logic Programs

Consider a set $\mathcal{A}$ of propositional atoms.

Definition

A **definite logic program** over $\mathcal{A}$ is a set P of rules of the form

$$a_0 \leftarrow a_1, \ldots, a_m$$

for $a_0, \ldots, a_m \in \mathcal{A}$ with $0 \leq m$.

A set of definite Horn clauses (exactly one positive literal).

Definition

- A set $S \subseteq \mathcal{A}$ is **closed** under a rule $a \leftarrow a_1, \ldots, a_m$ if and only if $\{a_1, \ldots, a_m\} \subseteq S$ implies $a \in S$.
- The **least model** of P is the $\subseteq$-least set that is closed under all rules in P.

Does such a least model always exist?

Semantics via Operators

Definition

Let P be a definite logic program over atoms $\mathcal{A}$.
The **one-step consequence operator** of P is given by ${}_PT : 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ with

$$S \mapsto \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m \in P,\ \{a_1, \ldots, a_m\} \subseteq S\}$$

The ${}_PT$ operator maps an interpretation S to a revised interpretation ${}_PT(S)$.

Semantics via Operators

Definition

Let P be a definite logic program over atoms $\mathcal{A}$.
The **one-step consequence operator** of P is given by ${}_P T : 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ with

$$S \mapsto \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m \in P,\ \{a_1, \ldots, a_m\} \subseteq S\}$$

The ${}_P T$ operator maps an interpretation S to a revised interpretation ${}_P T(S)$.

Proposition

For any definite logic program P, the operator ${}_P T$ is $\subseteq$-monotone.

Semantics via Operators

Definition

Let P be a definite logic program over atoms $\mathcal{A}$.
The **one-step consequence operator** of P is given by ${}_PT : 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ with

$$S \mapsto \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m \in P, \ \{a_1, \ldots, a_m\} \subseteq S\}$$

The ${}_PT$ operator maps an interpretation S to a revised interpretation ${}_PT(S)$.

Proposition

For any definite logic program P, the operator ${}_PT$ is $\subseteq$-monotone.

Proof.

Let $S_1 \subseteq S_2 \subseteq \mathcal{A}$ and $a \in {}_PT(S_1)$.

Semantics via Operators

Definition

Let P be a definite logic program over atoms $\mathcal{A}$.
The **one-step consequence operator** of P is given by ${}_PT : 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ with

$$S \mapsto \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m \in P, \ \{a_1, \ldots, a_m\} \subseteq S\}$$

The ${}_PT$ operator maps an interpretation S to a revised interpretation ${}_PT(S)$.

Proposition

For any definite logic program P, the operator ${}_PT$ is $\subseteq$-monotone.

Proof.

Let $S_1 \subseteq S_2 \subseteq \mathcal{A}$ and $a \in {}_PT(S_1)$.
Then there is a rule $a \leftarrow a_1, \ldots, a_m \in P$ with $\{a_1, \ldots, a_m\} \subseteq S_1$.

Semantics via Operators

Definition

Let P be a definite logic program over atoms $\mathcal{A}$.
The **one-step consequence operator** of P is given by ${}_PT : 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ with

$$S \mapsto \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m \in P, \ \{a_1, \ldots, a_m\} \subseteq S\}$$

The ${}_PT$ operator maps an interpretation S to a revised interpretation ${}_PT(S)$.

Proposition

For any definite logic program P, the operator ${}_PT$ is $\subseteq$-monotone.

Proof.

Let $S_1 \subseteq S_2 \subseteq \mathcal{A}$ and $a \in {}_PT(S_1)$.
Then there is a rule $a \leftarrow a_1, \ldots, a_m \in P$ with $\{a_1, \ldots, a_m\} \subseteq S_1$.
But then $\{a_1, \ldots, a_m\} \subseteq S_1 \subseteq S_2$, thus $a \in {}_PT(S_2)$. $\square$

Semantics via Operators

Definition

Let P be a definite logic program over atoms $\mathcal{A}$.
The **one-step consequence operator** of P is given by ${}_PT : 2^{\mathcal{A}} \to 2^{\mathcal{A}}$ with

$$S \mapsto \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m \in P, \ \{a_1, \ldots, a_m\} \subseteq S\}$$

The ${}_PT$ operator maps an interpretation S to a revised interpretation ${}_PT(S)$.

Proposition

For any definite logic program P, the operator ${}_PT$ is $\subseteq$-monotone.

Theorem

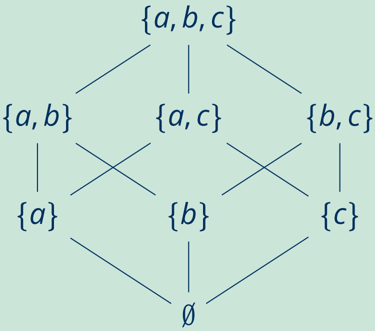
Every definite logic program P has a least model, given by the least fixpoint of ${}_PT$ in $(2^{\mathcal{A}}, \subseteq)$.

The least model of P captures its intended meaning.

Definite Logic Programs: Example

Example

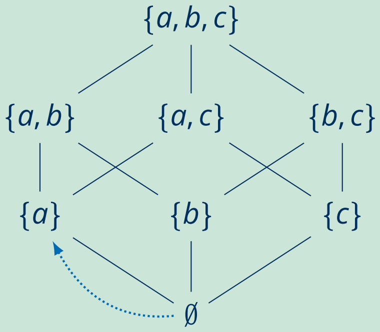
Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator ${}_PT$ maps as follows:



Definite Logic Programs: Example

Example

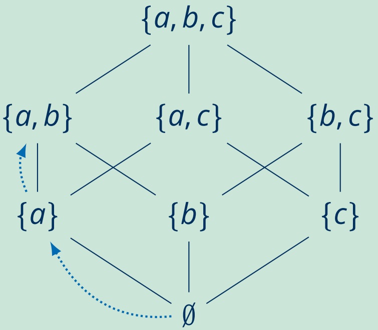
Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator ${}_PT$ maps as follows:



Definite Logic Programs: Example

Example

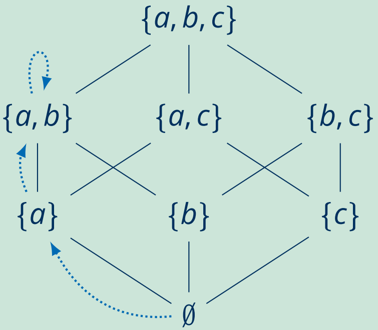
Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator ${}_PT$ maps as follows:



Definite Logic Programs: Example

Example

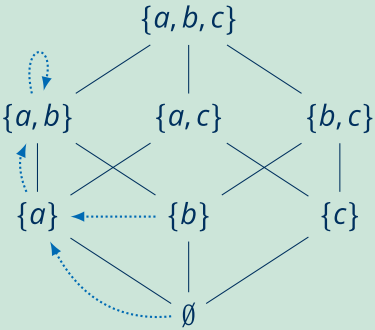
Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator $_PT$ maps as follows:



Definite Logic Programs: Example

Example

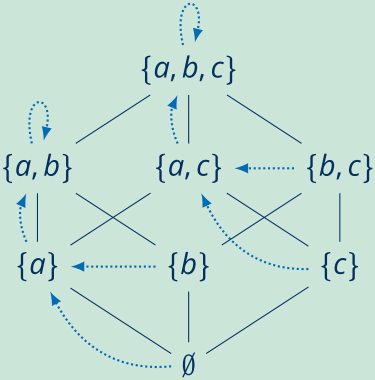
Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator ${}_PT$ maps as follows:



Definite Logic Programs: Example

Example

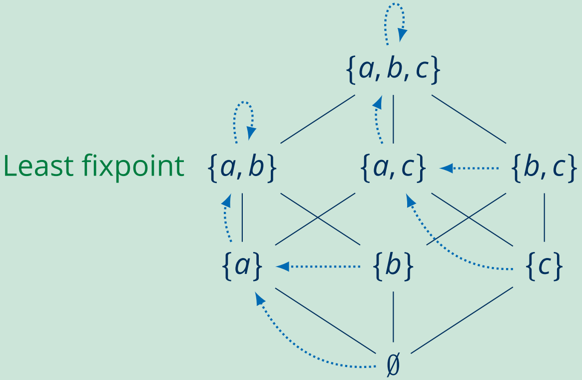
Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator ${}_PT$ maps as follows:



Definite Logic Programs: Example

Example

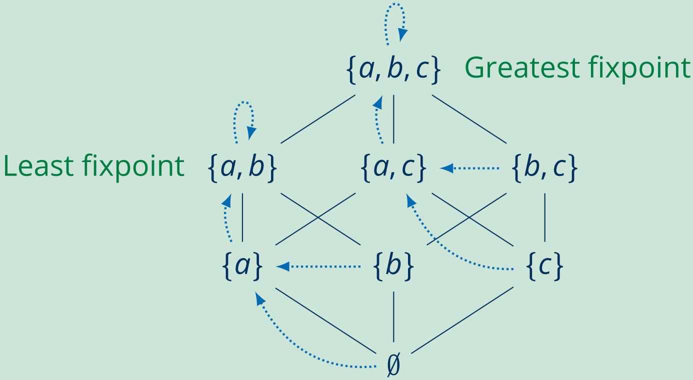
Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator $_PT$ maps as follows:



Definite Logic Programs: Example

Example

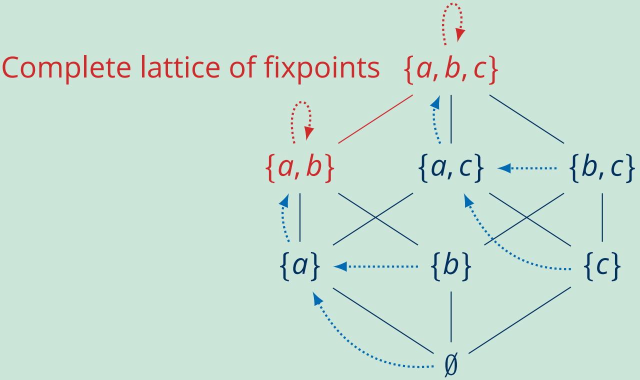
Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator ${}_PT$ maps as follows:



Definite Logic Programs: Example

Example

Consider $\mathcal{A} = \{a, b, c\}$ and the logic program $P = \{a \leftarrow, \quad b \leftarrow a, \quad c \leftarrow c\}$.
The operator ${}_PT$ maps as follows:



Normal Logic Programs

Definition

A **normal logic program** over $\mathcal{A}$ is a set P of rules of the form $a_0 \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, \sim a_n$ for $a_0, \ldots, a_n \in \mathcal{A}$ with $0 \leq m \leq n$.

Allow negated atoms $\sim a$ in rule bodies.

Normal Logic Programs

Definition

A **normal logic program** over $\mathcal{A}$ is a set P of rules of the form $a_0 \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n$ for $a_0, \ldots, a_n \in \mathcal{A}$ with $0 \leq m \leq n$.

Allow negated atoms ${\sim}a$ in rule bodies.

Definition

Let P be a normal logic program. The operator ${}_PT$ on $(2^{\mathcal{A}}, \subseteq)$ assigns thus:

$$S \mapsto \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n \in P, \\ \{a_1, \ldots, a_m\} \subseteq S,\ \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}$$

A set $S \subseteq \mathcal{A}$ is a **supported model** of P iff it is a fixpoint of ${}_PT$.

Allow to derive the rule head from S whenever the rule body is satisfied in S.

Alternative definition of supported models via Clark completion.

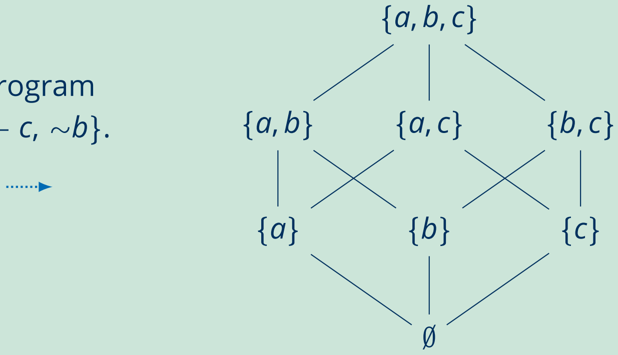
Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program

$P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}.$

Operator ${}_PT$ visualised by



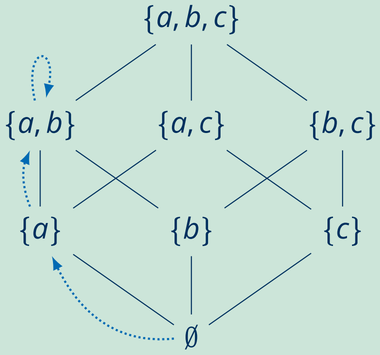
Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program

$P = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c, {\sim}b\}$.

Operator ${}_PT$ visualised by ┈┈▶

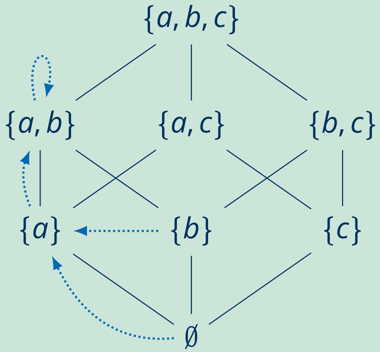


Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c, {\sim}b\}$.

Operator ${}_PT$ visualised by

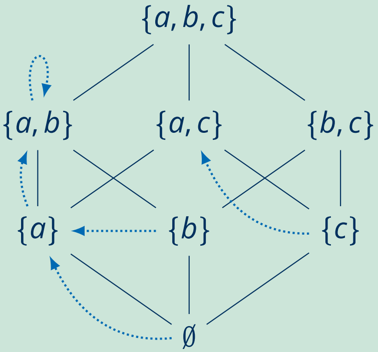


Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}$.

Operator ${}_PT$ visualised by ⋯⋯▸

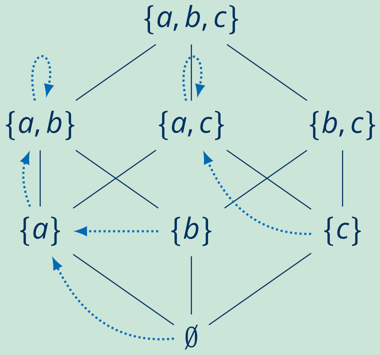


Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c, {\sim}b\}$.

Operator ${}_PT$ visualised by ⋯⋯▸

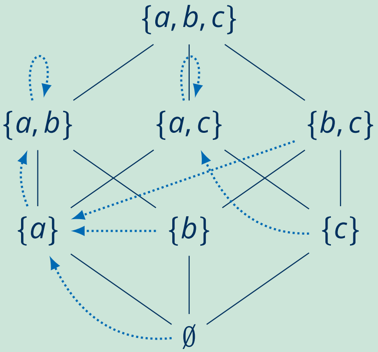


Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c, {\sim}b\}$.

Operator ${}_PT$ visualised by ········►

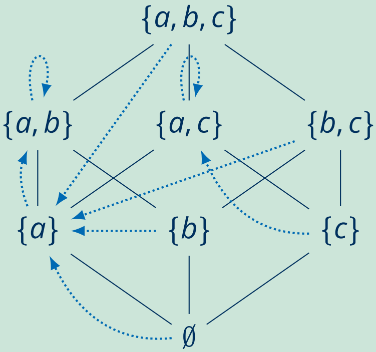


Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c, {\sim}b\}$.

Operator ${}_PT$ visualised by ┈┈▶

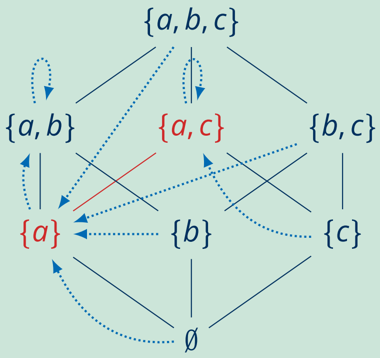


Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}$.

Operator ${}_P T$ visualised by ·······▸

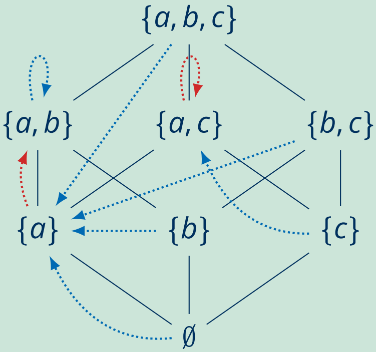


Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}$.

Operator ${}_PT$ visualised by ┈┈▶

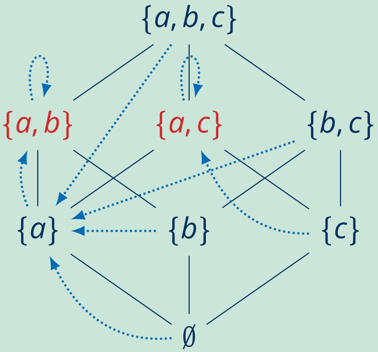


Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c, {\sim}b\}$.

Operator ${}_PT$ visualised by ⋯⋯▸



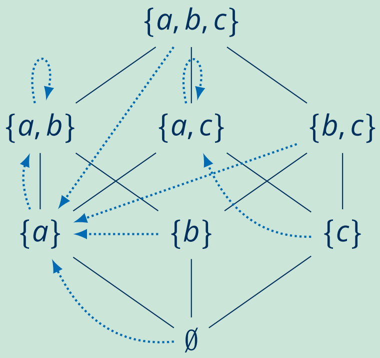
Normal Logic Programs: Example

Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}$.

Operator ${}_PT$ visualised by ⇢

${}_PT$ is not $\subseteq$-monotone.



Normal Logic Programs: Example

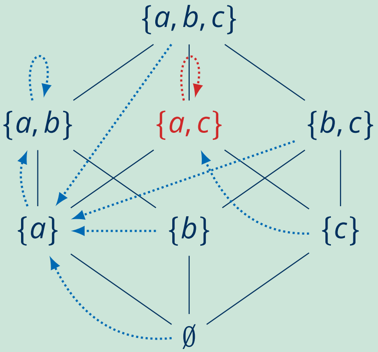
Example

Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program
$P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}$.

Operator ${}_PT$ visualised by ········►

${}_PT$ is not $\subseteq$-monotone.

In $\{a, c\}$, atom c justifies itself.



Normal Logic Programs: Example

Example

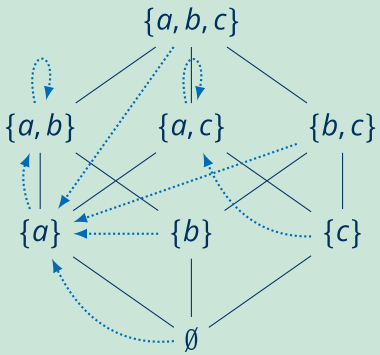
Let $\mathcal{A} = \{a, b, c\}$.
Consider the normal logic program

$P = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c, {\sim}b\}$.

Operator ${}_PT$ visualised by

${}_PT$ is not $\subseteq$-monotone.

In $\{a, c\}$, atom c justifies itself.



- How to avoid self-justification?
- How to obtain interpretation operators with “nice” properties?

Stable Model Semantics

Definition

Let P be a normal logic program and $S \subseteq \mathcal{A}$ be a set of atoms.
The **reduct of** P **with** S is the definite logic program P^S given by:

$$\{a \leftarrow a_1, \ldots, a_m \mid a \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n \in P, \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}$$

A set $S \subseteq \mathcal{A}$ is a **stable model of** P iff S is the $\subseteq$-least model of P^S.

Stable Model Semantics

Definition

Let P be a normal logic program and $S \subseteq \mathcal{A}$ be a set of atoms.
The **reduct of** P **with** S is the definite logic program P^S given by:

$$\{a \leftarrow a_1, \ldots, a_m \mid a \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, \sim a_n \in P, \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}$$

A set $S \subseteq \mathcal{A}$ is a **stable model of** P iff S is the $\subseteq$-least model of P^S.

In other words, P^S is obtained from P by:

- removing all rules containing $\sim a$ for some $a \in S$;
- removing all $\sim a$ from the remaining rules.

Stable Model Semantics

Definition

Let P be a normal logic program and $S \subseteq \mathcal{A}$ be a set of atoms.
The **reduct of** P **with** S is the definite logic program P^S given by:

$$\{a \leftarrow a_1, \ldots, a_m \mid a \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, \sim a_n \in P, \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}$$

A set $S \subseteq \mathcal{A}$ is a **stable model of** P iff S is the $\subseteq$-least model of P^S.

In other words, P^S is obtained from P by:

- removing all rules containing $\sim a$ for some $a \in S$;
- removing all $\sim a$ from the remaining rules.

Example (Continued.)

Reconsider logic program $P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}$ with supported models $\{a, b\}$ and $\{a, c\}$. Are they stable models?

Stable Model Semantics

Definition

Let P be a normal logic program and $S \subseteq \mathcal{A}$ be a set of atoms.
The **reduct of P with S** is the definite logic program P^S given by:

$$\{a \leftarrow a_1, \ldots, a_m \mid a \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, \sim a_n \in P, \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}$$

A set $S \subseteq \mathcal{A}$ is a **stable model of P** iff S is the $\subseteq$-least model of P^S.

In other words, P^S is obtained from P by:

- removing all rules containing $\sim a$ for some $a \in S$;
- removing all $\sim a$ from the remaining rules.

Example (Continued.)

Reconsider logic program $P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}$ with supported models $\{a, b\}$ and $\{a, c\}$. Are they stable models?

- $P^{\{a,b\}} = \{a \leftarrow, \quad b \leftarrow a\}$ with least model $\{a, b\}$, so $\{a, b\}$ is a stable model.

Stable Model Semantics

Definition

Let P be a normal logic program and $S \subseteq \mathcal{A}$ be a set of atoms.
The **reduct of P with S** is the definite logic program P^S given by:

$$\{a \leftarrow a_1, \ldots, a_m \mid a \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, \sim a_n \in P, \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}$$

A set $S \subseteq \mathcal{A}$ is a **stable model of P** iff S is the $\subseteq$-least model of P^S.

In other words, P^S is obtained from P by:

- removing all rules containing $\sim a$ for some $a \in S$;
- removing all $\sim a$ from the remaining rules.

Example (Continued.)

Reconsider logic program $P = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c, \sim b\}$ with supported models $\{a, b\}$ and $\{a, c\}$. Are they stable models?

- $P^{\{a,b\}} = \{a \leftarrow, \quad b \leftarrow a\}$ with least model $\{a, b\}$, so $\{a, b\}$ is a stable model.
- $P^{\{a,c\}} = \{a \leftarrow, \quad c \leftarrow c\}$ with least model $\{a\}$, so $\{a, c\}$ is not stable.

Stocktaking

- Monotone operators in complete lattices have (least and greatest) fixpoints.
- Operators can be associated with knowledge bases such that their fixpoints correspond to models.
- Definite logic programs lead to an operator that is monotone on $(2^{\mathcal{A}}, \subseteq)$, and thus have unique least models.
- Normal logic programs lead to a non-monotone operator; model existence and uniqueness cannot be guaranteed.
- Stable model semantics deals with self-justification.
- Can we find an operator-based version of stable model semantics?

Approximating Operators

Approximating Operators

Main Idea

Use a more fine-grained structure to keep track of (partial) truth values.

Desiderata

- Preserve "interpretation revision" character of operators
- Preserve correspondence of fixpoints with models
- Obtain useful properties of operators

Approach

- Approximate sets of models by intervals.
- Use an information ordering on these approximations.
- Approximate operators by approximators – operators on intervals.
- Guarantee that fixpoints of approximators contain original fixpoints.

From Lattices to Bilattices

Definition

Let $(L, \leqslant)$ be a partially ordered set.
Its associated **information bilattice** is $(L^2, \leq_i)$ with $L^2 = L \times L$ and

$$(u, v) \leq_i (x, y) \quad \text{iff} \quad u \leqslant x \text{ and } y \leqslant v$$

- A pair (x, y) approximates all $z \in L$ with $x \leqslant z \leqslant y$.
- Information ordering $\triangleq$ interval inclusion: $(u, v) \leq_i (x, y)$ iff $[x, y] \subseteq [u, v]$

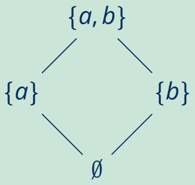
Proposition

If $(L, \leqslant)$ is a complete lattice, then $(L^2, \leq_i)$ is a complete lattice. For $S \subseteq L^2$:

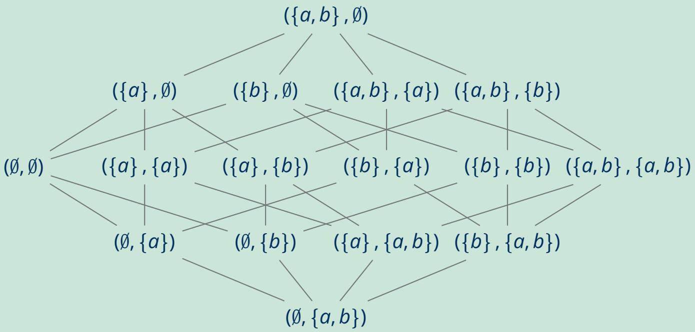
$$\bigwedge_i S = \left(\bigwedge S_1, \bigvee S_2\right) \quad \text{and} \quad \bigvee_i S = \left(\bigvee S_1, \bigwedge S_2\right) \qquad \begin{array}{l} S_1 = \{x \mid (x,y) \in S\} \\ S_2 = \{y \mid (x,y) \in S\} \end{array}$$

From Lattice to Bilattice: Example

Example



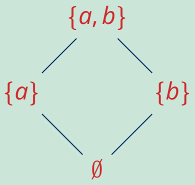
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

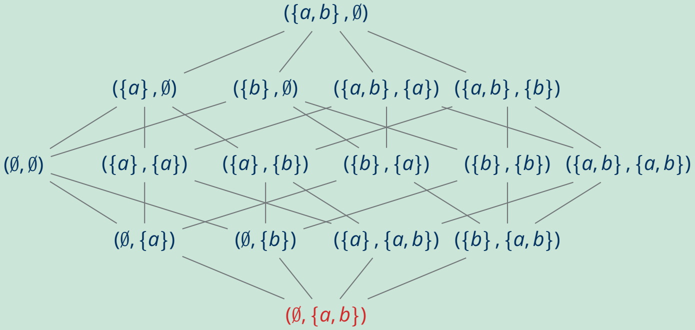


Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

From Lattice to Bilattice: Example

Example

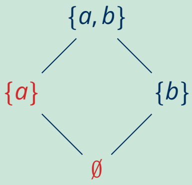




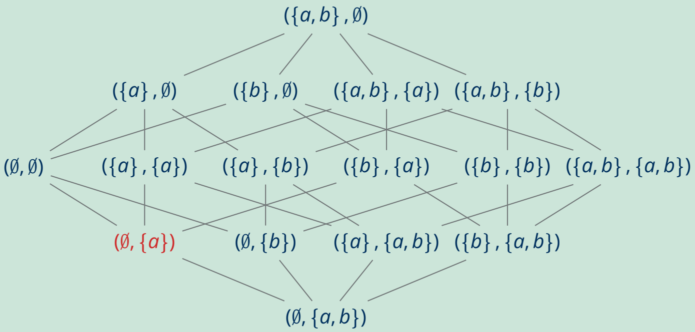
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$ Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

From Lattice to Bilattice: Example

Example



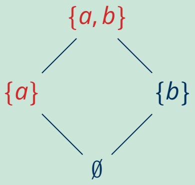
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$



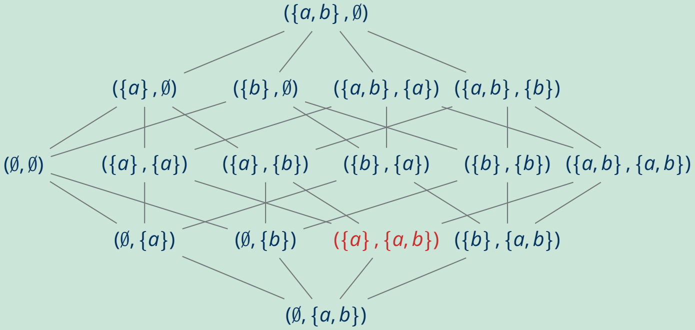
Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

From Lattice to Bilattice: Example

Example



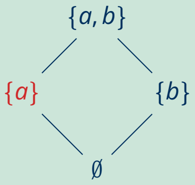
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

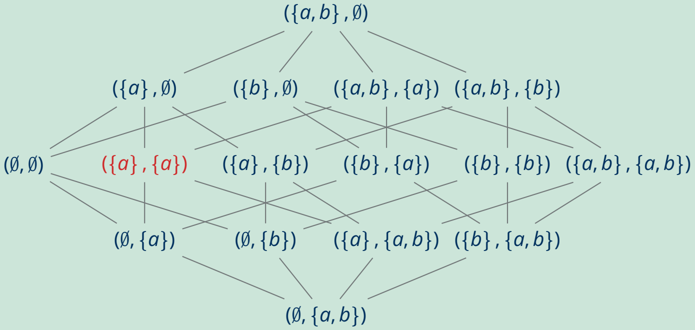


Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

From Lattice to Bilattice: Example

Example



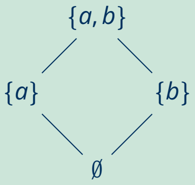


Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

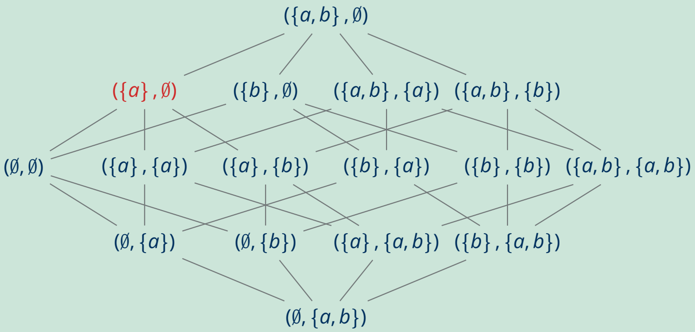
Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

From Lattice to Bilattice: Example

Example



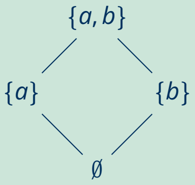
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

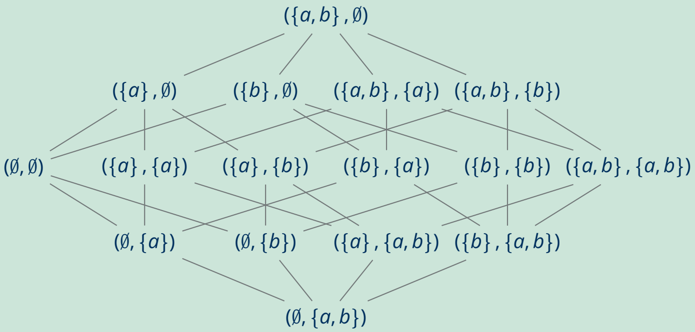


Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

From Lattice to Bilattice: Example

Example





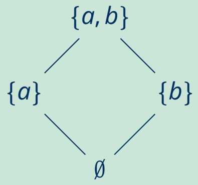
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

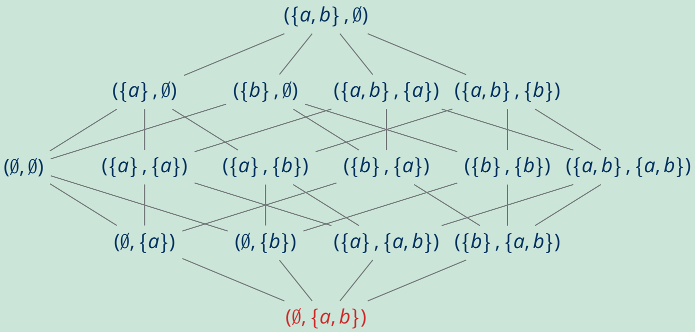
Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

Pairs in the bilattice correspond to four-valued interpretations $v : \{a, b\} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

From Lattice to Bilattice: Example

Example





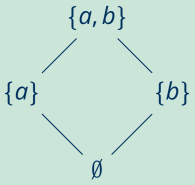
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$ Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

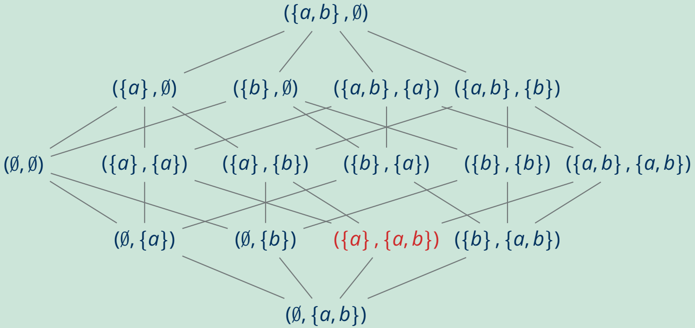
Pairs in the bilattice correspond to four-valued interpretations $v : \{a, b\} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

$$\{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}\}$$

From Lattice to Bilattice: Example

Example





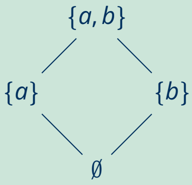
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$ Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

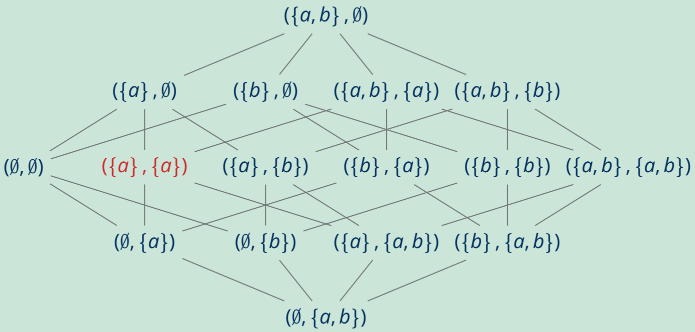
Pairs in the bilattice correspond to four-valued interpretations $v : \{a, b\} \rightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

$$\{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}\}$$

From Lattice to Bilattice: Example

Example





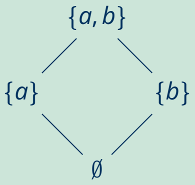
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$ Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

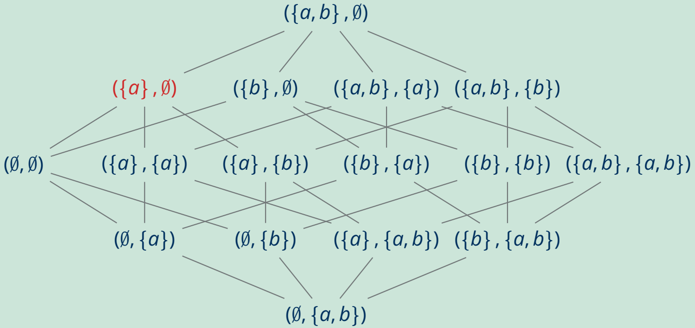
Pairs in the bilattice correspond to four-valued interpretations $v : \{a, b\} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

$$\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}\}$$

From Lattice to Bilattice: Example

Example





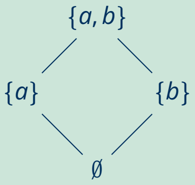
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$ Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

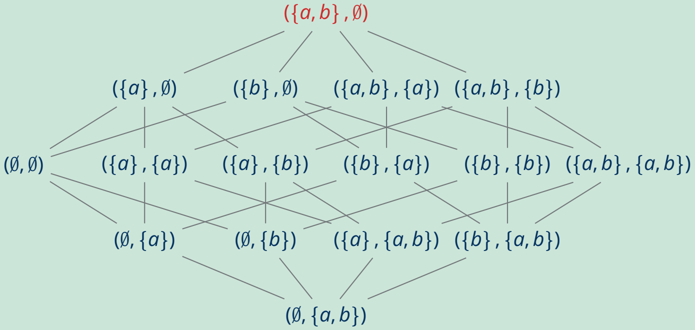
Pairs in the bilattice correspond to four-valued interpretations $v : \{a, b\} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

$$\{a \mapsto \mathbf{i}, b \mapsto \mathbf{f}\}$$

From Lattice to Bilattice: Example

Example





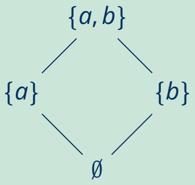
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$ Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

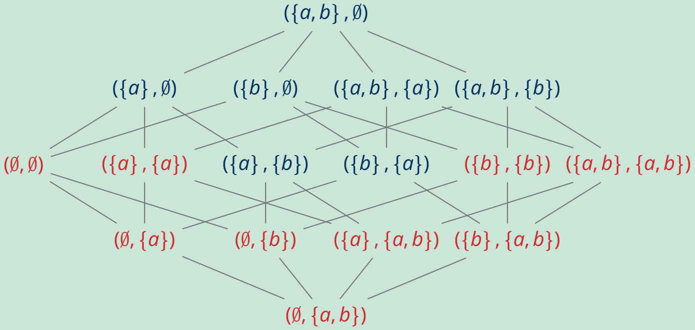
Pairs in the bilattice correspond to four-valued interpretations $v : \{a,b\} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

$$\{a \mapsto \mathbf{i}, b \mapsto \mathbf{i}\}$$

From Lattice to Bilattice: Example

Example





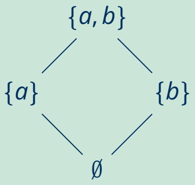
Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$ Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

Pairs in the bilattice correspond to four-valued interpretations $v : \{a, b\} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

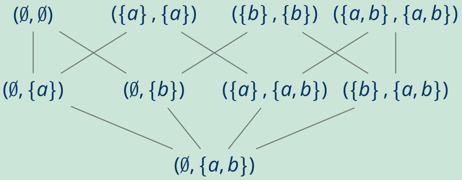
We will mostly be concerned with the consistent pairs.

From Lattice to Bilattice: Example

Example



Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$



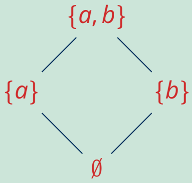
Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

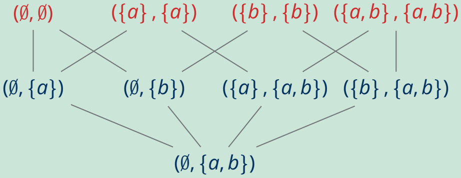
Pairs in the bilattice correspond to four-valued interpretations $v : \{a, b\} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

We will mostly be concerned with the consistent pairs.

From Lattice to Bilattice: Example

Example





Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$ Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

Pairs in the bilattice correspond to four-valued interpretations $v : \{a, b\} \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}, \mathbf{i}\}$.

Elements of the original lattice correspond to exact pairs.

Approximator

Recall approach: Approximate lattice operators on a richer structure.

Definition

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be an operator.
An operator $A : L^2 \to L^2$ **approximates** O iff for all $x \in L$, we have

$$A(x, x) = (O(x), O(x))$$

A is an **approximator** iff A approximates some O and A is $\leq_i$-monotone.

Approximator coincides with the operator on exact pairs.

Approximator

Recall approach: Approximate lattice operators on a richer structure.

Definition

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be an operator.
An operator $A : L^2 \to L^2$ **approximates** O iff for all $x \in L$, we have

$$A(x, x) = (O(x), O(x))$$

A is an **approximator** iff A approximates some O and A is $\leq_i$-monotone.

Approximator coincides with the operator on exact pairs.

$A : L^2 \to L^2$ induces $A_1, A_2 : L^2 \to L$ with $A(x, y) = (A_1(x, y), A_2(x, y))$.

Approximator

Recall approach: Approximate lattice operators on a richer structure.

Definition

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be an operator.
An operator $A : L^2 \to L^2$ **approximates** O iff for all $x \in L$, we have

$$A(x,x) = (O(x), O(x))$$

A is an **approximator** iff A approximates some O and A is $\leq_i$-monotone.

Approximator coincides with the operator on exact pairs.

$A : L^2 \to L^2$ induces $A_1, A_2 : L^2 \to L$ with $A(x,y) = (A_1(x,y), A_2(x,y))$.

Definition

An approximator is **symmetric** iff $A_1(x,y) = A_2(y,x)$.

Approximator

Recall approach: Approximate lattice operators on a richer structure.

Definition

Let $(L, \leqslant)$ be a complete lattice and $O : L \to L$ be an operator.
An operator $A : L^2 \to L^2$ **approximates** O iff for all $x \in L$, we have

$$A(x, x) = (O(x), O(x))$$

A is an **approximator** iff A approximates some O and A is $\leq_i$-monotone.

Approximator coincides with the operator on exact pairs.

$A : L^2 \to L^2$ induces $A_1, A_2 : L^2 \to L$ with $A(x, y) = (A_1(x, y), A_2(x, y))$.

Definition

An approximator is **symmetric** iff $A_1(x, y) = A_2(y, x)$.

If A is symmetric, then $A(x, y) = (A_1(x, y), A_1(y, x))$, so A_1 fully specifies A.

Approximator: Example

Example

Let P be a normal logic program.
Recall its one-step consequence operator ${}_PT$, defined by

$$
{}_PT(S) = \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n \in P, \\
\{a_1, \ldots, a_m\} \subseteq S, \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}
$$

Approximator: Example

Example

Let P be a normal logic program.
Recall its one-step consequence operator ${}_PT$, defined by

$$ {}_PT(S) = \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, \sim a_n \in P, \\ \{a_1, \ldots, a_m\} \subseteq S, \ \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\} $$

A symmetric approximator for ${}_PT$ is given by ${}_P\mathcal{T}$ with

$$ {}_P\mathcal{T}_1(L, U) = \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, \sim a_n \in P, \\ \{a_1, \ldots, a_m\} \subseteq L, \ \{a_{m+1}, \ldots, a_n\} \cap U = \emptyset\} $$

Approximator: Example

Example

Let P be a normal logic program.
Recall its one-step consequence operator ${}_PT$, defined by

$$
{}_PT(S) = \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n \in P, \\
\{a_1, \ldots, a_m\} \subseteq S, \ \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}
$$

A symmetric approximator for ${}_PT$ is given by ${}_P\mathcal{T}$ with

$$
{}_P\mathcal{T}_1(L, U) = \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n \in P, \\
\{a_1, \ldots, a_m\} \subseteq L, \ \{a_{m+1}, \ldots, a_n\} \cap U = \emptyset\}
$$

That is, ${}_P\mathcal{T}(L, U) = \left({}_P\mathcal{T}_1(L, U), {}_P\mathcal{T}_1(U, L)\right)$.

Approximator: Example

Example

Let P be a normal logic program.
Recall its one-step consequence operator ${}_PT$, defined by

$$
{}_PT(S) = \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n \in P, \\
\{a_1, \ldots, a_m\} \subseteq S, \ \{a_{m+1}, \ldots, a_n\} \cap S = \emptyset\}
$$

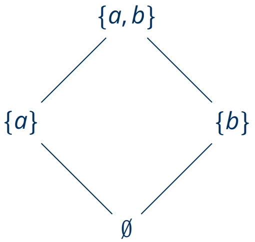
A symmetric approximator for ${}_PT$ is given by ${}_P\mathcal{T}$ with

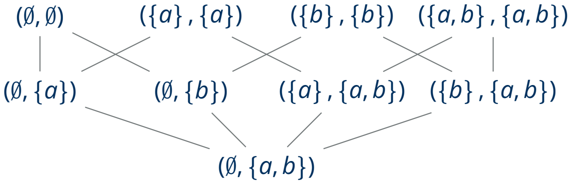
$$
{}_P\mathcal{T}_1(L, U) = \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n \in P, \\
\{a_1, \ldots, a_m\} \subseteq L, \ \{a_{m+1}, \ldots, a_n\} \cap U = \emptyset\}
$$

That is, ${}_P\mathcal{T}(L, U) = \left({}_P\mathcal{T}_1(L, U), {}_P\mathcal{T}_1(U, L)\right)$.

For new lower bound: check truth against lower, falsity against upper bound.

Approximator ${}_P\mathcal{T}$: Example





Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

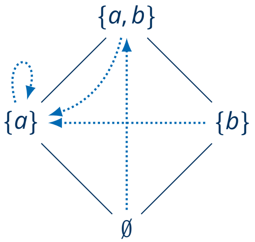
Normal logic program
$P = \{a \leftarrow, \quad b \leftarrow \sim a, \sim b\}$

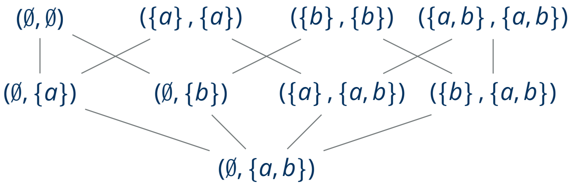
${}_PT$: ········►

Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

Approximator ${}_P\mathcal{T}$ for ${}_PT$: - - -►

Approximator $_P\mathcal{T}$: Example





Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

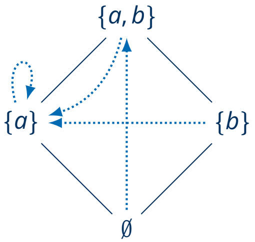
Normal logic program
$P = \{a \leftarrow, \quad b \leftarrow \sim a, \sim b\}$

$_PT$: ·········►

Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

Approximator $_P\mathcal{T}$ for $_PT$: - - -►

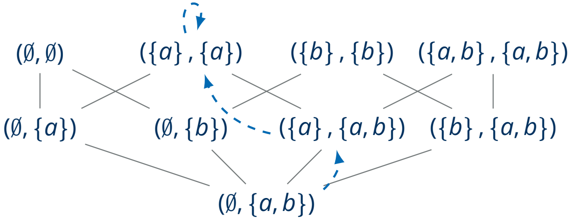
Approximator $_P\mathcal{T}$: Example



Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

Normal logic program
$P = \{a \leftarrow, \quad b \leftarrow {\sim}a, {\sim}b\}$

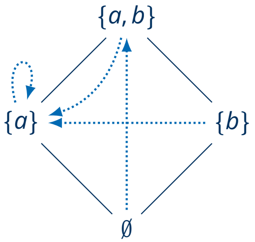
$_PT$: (dotted arrow)

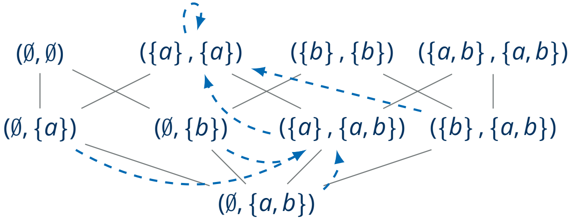


Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

Approximator $_P\mathcal{T}$ for $_PT$: (dashed arrow)

Approximator $_P\mathcal{T}$: Example





Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

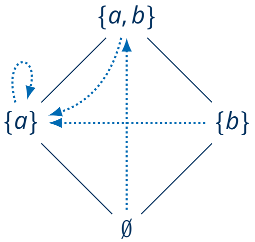
Normal logic program
$P = \{a \leftarrow, \quad b \leftarrow {\sim}a, {\sim}b\}$

$_PT$: ········►

Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

Approximator $_P\mathcal{T}$ for $_PT$: - - - ►

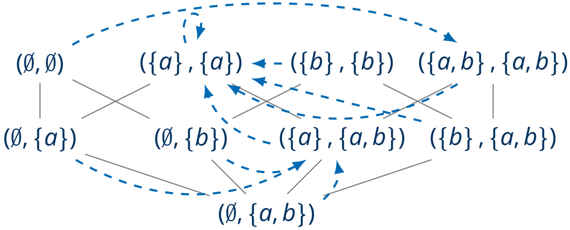
Approximator $_P\mathcal{T}$: Example



Original lattice $\left(2^{\{a,b\}}, \subseteq\right)$

Normal logic program
$P = \{a \leftarrow, \quad b \leftarrow \sim a, \sim b\}$

$_PT$: ·········►



Bilattice $\left(2^{\{a,b\}} \times 2^{\{a,b\}}, \leq_i\right)$

Approximator $_P\mathcal{T}$ for $_PT$: - - -►

Quiz: Approximator $_P\mathcal{T}$

Recall that for $L, U \subseteq \mathcal{A}$ we defined $_P\mathcal{T}(L, U) = ({_P\mathcal{T}_1}(L, U), {_P\mathcal{T}_1}(U, L))$ with

$$
{_P\mathcal{T}_1}(L, U) = \{a_0 \in \mathcal{A} \mid a_0 \leftarrow a_1, \ldots, a_m, {\sim}a_{m+1}, \ldots, {\sim}a_n \in P, \\ \{a_1, \ldots, a_m\} \subseteq L, \ \{a_{m+1}, \ldots, a_n\} \cap U = \emptyset\}
$$

Quiz

Consider the normal logic program P: …

Approximator: Observations (1)

Lemma

Let $(L, \leqslant)$ be a complete lattice and A an approximator on $(L^2, \leq_i)$.

1. If C is a non-empty chain of consistent pairs, then $\bigvee_i C$ is consistent.
2. If (x, y) is consistent, then $A(x, y)$ is consistent.

Approximators map consistent pairs to consistent pairs.

Approximator: Observations (1)

Lemma

Let $(L, \leqslant)$ be a complete lattice and A an approximator on $(L^2, \leq_i)$.

1. If C is a non-empty chain of consistent pairs, then $\bigvee_i C$ is consistent.
2. If (x, y) is consistent, then $A(x, y)$ is consistent.

Approximators map consistent pairs to consistent pairs.

Proof.

1. Let $a, b \in C$. Since C is a chain, $a \leq_i b$ (then $a_1 \leqslant b_1 \leqslant b_2$) or $b \leq_i a$ (then $a_1 \leqslant a_2 \leqslant b_2$).

Approximator: Observations (1)

Lemma

Let $(L, \leqslant)$ be a complete lattice and A an approximator on $(L^2, \leq_i)$.

1. If C is a non-empty chain of consistent pairs, then $\bigvee_i C$ is consistent.
2. If (x, y) is consistent, then $A(x, y)$ is consistent.

Approximators map consistent pairs to consistent pairs.

Proof.

1. Let $a, b \in C$. Since C is a chain, $a \leq_i b$ (then $a_1 \leqslant b_1 \leqslant b_2$) or $b \leq_i a$ (then $a_1 \leqslant a_2 \leqslant b_2$). In any case, $a_1 \leqslant b_2$. So every $c_2 \in C_2$ is an upper bound of C_1, and $\bigvee C_1 \leqslant c_2$.

Approximator: Observations (1)

Lemma

Let $(L, \leqslant)$ be a complete lattice and A an approximator on $(L^2, \leq_i)$.

1. If C is a non-empty chain of consistent pairs, then $\bigvee_i C$ is consistent.
2. If (x, y) is consistent, then $A(x, y)$ is consistent.

Approximators map consistent pairs to consistent pairs.

Proof.

1. Let $a, b \in C$. Since C is a chain, $a \leq_i b$ (then $a_1 \leqslant b_1 \leqslant b_2$) or $b \leq_i a$ (then $a_1 \leqslant a_2 \leqslant b_2$). In any case, $a_1 \leqslant b_2$. So every $c_2 \in C_2$ is an upper bound of C_1, and $\bigvee C_1 \leqslant c_2$. Hence $\bigvee C_1$ is a lower bound of C_2 and $\bigvee C_1 \leqslant \bigwedge C_2$.

Approximator: Observations (1)

Lemma

Let $(L, \leqslant)$ be a complete lattice and A an approximator on $(L^2, \leq_i)$.

1. If C is a non-empty chain of consistent pairs, then $\bigvee_i C$ is consistent.
2. If (x, y) is consistent, then $A(x, y)$ is consistent.

Approximators map consistent pairs to consistent pairs.

Proof.

1. Let $a, b \in C$. Since C is a chain, $a \leq_i b$ (then $a_1 \leqslant b_1 \leqslant b_2$) or $b \leq_i a$ (then $a_1 \leqslant a_2 \leqslant b_2$). In any case, $a_1 \leqslant b_2$. So every $c_2 \in C_2$ is an upper bound of C_1, and $\bigvee C_1 \leqslant c_2$. Hence $\bigvee C_1$ is a lower bound of C_2 and $\bigvee C_1 \leqslant \bigwedge C_2$.
2. If $x \leqslant y$, then for z with $x \leqslant z \leqslant y$ we have $(x, y) \leq_i (z, z)$. A is $\leq_i$-monotone, thus $A(x, y) \leq_i A(z, z)$.

Approximator: Observations (1)

Lemma

Let $(L, \leqslant)$ be a complete lattice and A an approximator on $(L^2, \leq_i)$.

1. If C is a non-empty chain of consistent pairs, then $\bigvee_i C$ is consistent.
2. If (x, y) is consistent, then $A(x, y)$ is consistent.

Approximators map consistent pairs to consistent pairs.

Proof.

1. Let $a, b \in C$. Since C is a chain, $a \leq_i b$ (then $a_1 \leqslant b_1 \leqslant b_2$) or $b \leq_i a$ (then $a_1 \leqslant a_2 \leqslant b_2$). In any case, $a_1 \leqslant b_2$. So every $c_2 \in C_2$ is an upper bound of C_1, and $\bigvee C_1 \leqslant c_2$. Hence $\bigvee C_1$ is a lower bound of C_2 and $\bigvee C_1 \leqslant \bigwedge C_2$.
2. If $x \leqslant y$, then for z with $x \leqslant z \leqslant y$ we have $(x, y) \leq_i (z, z)$. A is $\leq_i$-monotone, thus $A(x, y) \leq_i A(z, z)$. A approximates some O, thus $A(z, z) = (O(z), O(z))$.

Approximator: Observations (1)

Lemma

Let $(L, \leqslant)$ be a complete lattice and A an approximator on $(L^2, \leq_i)$.

1. If C is a non-empty chain of consistent pairs, then $\bigvee_i C$ is consistent.
2. If (x, y) is consistent, then $A(x, y)$ is consistent.

Approximators map consistent pairs to consistent pairs.

Proof.

1. Let $a, b \in C$. Since C is a chain, $a \leq_i b$ (then $a_1 \leqslant b_1 \leqslant b_2$) or $b \leq_i a$ (then $a_1 \leqslant a_2 \leqslant b_2$). In any case, $a_1 \leqslant b_2$. So every $c_2 \in C_2$ is an upper bound of C_1, and $\bigvee C_1 \leqslant c_2$. Hence $\bigvee C_1$ is a lower bound of C_2 and $\bigvee C_1 \leqslant \bigwedge C_2$.
2. If $x \leqslant y$, then for z with $x \leqslant z \leqslant y$ we have $(x, y) \leq_i (z, z)$. A is $\leq_i$-monotone, thus $A(x, y) \leq_i A(z, z)$. A approximates some O, thus $A(z, z) = (O(z), O(z))$. In combination $A_1(x, y) \leqslant O(z) \leqslant A_2(x, y)$.

□

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

The least fixpoint of A is consistent and approximates all fixpoints of O.

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

The least fixpoint of A is consistent and approximates all fixpoints of O.

Proof.

1. By Knaster/Tarski, A has a $\leq_i$-least fixpoint (x^*, y^*).

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

The least fixpoint of A is consistent and approximates all fixpoints of O.

Proof.

1. By Knaster/Tarski, A has a $\leq_i$-least fixpoint (x^*, y^*). It is also consistent:

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

The least fixpoint of A is consistent and approximates all fixpoints of O.

Proof.

1. By Knaster/Tarski, A has a $\leq_i$-least fixpoint (x^*, y^*). It is also consistent: Define $Q = \left\{(x,y) \in L^2 \;\middle|\; x \leqslant y \;\;\&\;\; (x,y) \leq_i A(x,y) \;\;\&\;\; (x,y) \leq_i (x^*, y^*)\right\}$.

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

The least fixpoint of A is consistent and approximates all fixpoints of O.

Proof.

1. By Knaster/Tarski, A has a $\leq_i$-least fixpoint (x^*, y^*). It is also consistent:
   Define $Q = \left\{(x,y) \in L^2 \;\middle|\; x \leqslant y \;\;\&\;\; (x,y) \leq_i A(x,y) \;\;\&\;\; (x,y) \leq_i (x^*,y^*)\right\}$.
   Q is non-empty as $(\bot, \top) \in Q$.

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

The least fixpoint of A is consistent and approximates all fixpoints of O.

Proof.

1. By Knaster/Tarski, A has a $\leq_i$-least fixpoint (x^*, y^*). It is also consistent: Define $Q = \left\{(x,y) \in L^2 \,\middle|\, x \leqslant y \;\;\&\;\; (x,y) \leq_i A(x,y) \;\;\&\;\; (x,y) \leq_i (x^*, y^*)\right\}$. Q is non-empty as $(\bot, \top) \in Q$. Each non-empty chain in Q has an upper bound in Q,

Let $\emptyset \neq C \subseteq Q$ be a chain. Define $d = \bigvee_i C$. (1) By the previous lemma, d is consistent. (2) For every $c \in C$ we have $c \leq_i d$ and thus $c \leq_i A(c) \leq_i A(d)$; thus $A(d)$ is an upper bound of C, whence $d \leq_i A(d)$. (3) We know that $C \subseteq Q$ whence (x^*, y^*) is an upper bound of C, thus $d \leq_i (x^*, y^*)$.

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

The least fixpoint of A is consistent and approximates all fixpoints of O.

Proof.

1. By Knaster/Tarski, A has a $\leq_i$-least fixpoint (x^*, y^*). It is also consistent: Define $Q = \left\{(x,y) \in L^2 \;\middle|\; x \leqslant y \;\;\&\;\; (x,y) \leq_i A(x,y) \;\;\&\;\; (x,y) \leq_i (x^*, y^*)\right\}$. Q is non-empty as $(\bot, \top) \in Q$. Each non-empty chain in Q has an upper bound in Q, therefore by Zorn's Lemma, Q has a maximal element, p.

Let $\emptyset \neq C \subseteq Q$ be a chain. Define $d = \bigvee_i C$. (1) By the previous lemma, d is consistent. (2) For every $c \in C$ we have $c \leq_i d$ and thus $c \leq_i A(c) \leq_i A(d)$; thus $A(d)$ is an upper bound of C, whence $d \leq_i A(d)$. (3) We know that $C \subseteq Q$ whence (x^*, y^*) is an upper bound of C, thus $d \leq_i (x^*, y^*)$.

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

The least fixpoint of A is consistent and approximates all fixpoints of O.

Proof.

1. By Knaster/Tarski, A has a $\leq_i$-least fixpoint (x^*, y^*). It is also consistent: Define $Q = \left\{(x,y) \in L^2 \;\middle|\; x \leqslant y \;\;\&\;\; (x,y) \leq_i A(x,y) \;\;\&\;\; (x,y) \leq_i (x^*, y^*)\right\}$. Q is non-empty as $(\bot, \top) \in Q$. Each non-empty chain in Q has an upper bound in Q, therefore by Zorn's Lemma, Q has a maximal element, ρ. Since ρ is maximal, $\rho \leq_i A(\rho)$ directly yields $A(\rho) = \rho = (x^*, y^*)$.

Let $\emptyset \neq C \subseteq Q$ be a chain. Define $d = \bigvee_i C$. (1) By the previous lemma, d is consistent. (2) For every $c \in C$ we have $c \leq_i d$ and thus $c \leq_i A(c) \leq_i A(d)$; thus $A(d)$ is an upper bound of C, whence $d \leq_i A(d)$. (3) We know that $C \subseteq Q$ whence (x^*, y^*) is an upper bound of C, thus $d \leq_i (x^*, y^*)$.

Approximator: Observations (2)

Theorem

Let $(L, \leqslant)$ be a complete lattice with $O : L \to L$, and A an approximator for O.

1. A has a $\leq_i$-least fixpoint (x^*, y^*) with $x^* \leqslant y^*$.
2. Every fixpoint z of O satisfies $x^* \leqslant z \leqslant y^*$.

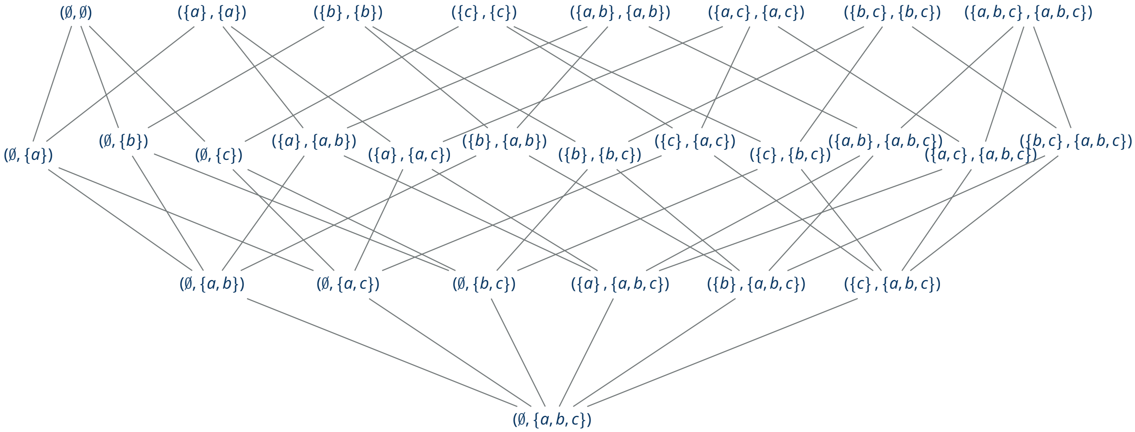
The least fixpoint of A is consistent and approximates all fixpoints of O.

Proof.

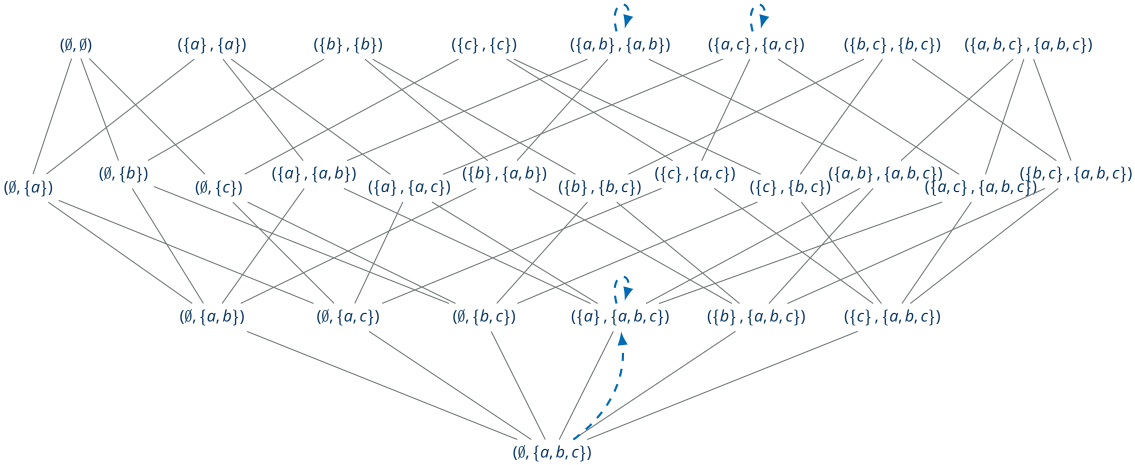
1. By Knaster/Tarski, A has a $\leq_i$-least fixpoint (x^*, y^*). It is also consistent: Define $Q = \left\{(x,y) \in L^2 \;\middle|\; x \leqslant y \;\;\&\;\; (x,y) \leq_i A(x,y) \;\;\&\;\; (x,y) \leq_i (x^*,y^*)\right\}$. Q is non-empty as $(\bot, \top) \in Q$. Each non-empty chain in Q has an upper bound in Q, therefore by Zorn's Lemma, Q has a maximal element, p. Since p is maximal, $p \leq_i A(p)$ directly yields $A(p) = p = (x^*, y^*)$.
2. If $O(z) = z$ then $A(z,z) = (O(z), O(z)) = (z,z)$ and $(x^*, y^*) \leq_i (z,z)$. □

Let $\emptyset \neq C \subseteq Q$ be a chain. Define $d = \bigvee_i C$. (1) By the previous lemma, d is consistent. (2) For every $c \in C$ we have $c \leq_i d$ and thus $c \leq_i A(c) \leq_i A(d)$; thus $A(d)$ is an upper bound of C, whence $d \leq_i A(d)$. (3) We know that $C \subseteq Q$ whence (x^*, y^*) is an upper bound of C, thus $d \leq_i (x^*, y^*)$.

Approximator $_P\mathcal{T}$: Examples

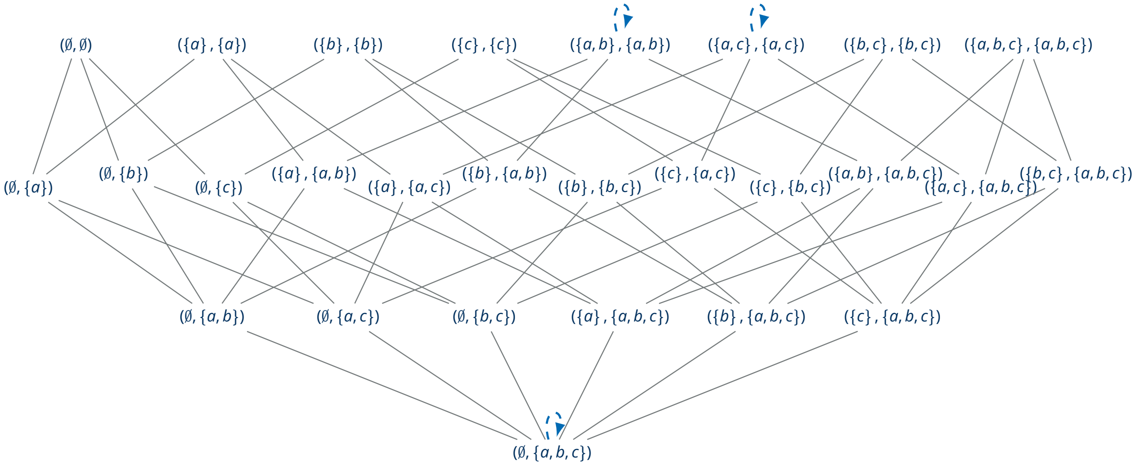


Approximator $_P\mathcal{T}$: Examples



$$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$$

Approximator $_P\mathcal{T}$: Examples



$$P_2 = \{a \leftarrow b, \quad a \leftarrow c, \quad b \leftarrow \sim c, \quad c \leftarrow \sim b\}$$

Recovering Semantics

Approximator fixpoints give rise to several semantics.

Proposition

Let P be a normal logic program over $\mathcal{A}$ with approximator ${}_P\mathfrak{T}$, $X \subseteq Y \subseteq \mathcal{A}$.

- X is a supported model of P iff ${}_P\mathfrak{T}(X, X) = (X, X)$.
- (X, Y) is a three-valued supported model of P iff ${}_P\mathfrak{T}(X, Y) = (X, Y)$.
- (X, Y) is the Kripke-Kleene semantics of P iff $(X, Y) = \text{lfp}({}_P\mathfrak{T})$.

But what about stable model semantics?

Stable Operators

Stable Operator: Intuition

The Gelfond-Lifschitz Reduct of P …

- …starts out with a two-valued interpretation $S \subseteq \mathcal{A}$;
- …removes all rules requiring some $a \in S$ to be false;
- …assumes all $a \in \mathcal{A} \setminus S$ to be false in the remaining rules.

Stable Operator: Intuition

The Gelfond-Lifschitz Reduct of P …

- …starts out with a two-valued interpretation $S \subseteq \mathcal{A}$;
- …removes all rules requiring some $a \in S$ to be false;
- …assumes all $a \in \mathcal{A} \setminus S$ to be false in the remaining rules.

- To obtain reduct P^S, assume all and only atoms $a \in \mathcal{A} \setminus S$ to be false.
- Using P^S, try to constructively prove all and only atoms $a \in S$ to be true.
- P^S is a definite logic program, so ${}_{P^S}T$ is a $\subseteq$-monotone operator on $(2^{\mathcal{A}}, \subseteq)$.

Stable Operator: Intuition

The Gelfond-Lifschitz Reduct of P ...

- ...starts out with a two-valued interpretation $S \subseteq \mathcal{A}$;
- ...removes all rules requiring some $a \in S$ to be false;
- ...assumes all $a \in \mathcal{A} \setminus S$ to be false in the remaining rules.

- To obtain reduct P^S, assume all and only atoms $a \in \mathcal{A} \setminus S$ to be false.
- Using P^S, try to constructively prove all and only atoms $a \in S$ to be true.
- P^S is a definite logic program, so ${}_{P^S}T$ is a $\subseteq$-monotone operator on $(2^{\mathcal{A}}, \subseteq)$.

Expressing the Reduct via an Operator

- For pair (X, Y), an $a \in \mathcal{A}$ is true iff $a \in X$; atom a is false iff $a \notin Y$.
- Use ${}_{P}\mathcal{T}_1$ to reconstruct what is true, fixing the upper bound to S:

$${}_{P}\mathcal{T}_1(\cdot, S) : 2^{\mathcal{A}} \to 2^{\mathcal{A}}, \quad X \mapsto {}_{P}\mathcal{T}_1(X, S)$$

Stable Operator: Preparation

Proposition

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.
For every pair $(x, y) \in L^2$, the following operators are $\leqslant$-monotone:

$$A_1(\cdot, y) : L \to L, \quad z \mapsto A_1(z, y) \qquad \text{and} \qquad A_2(x, \cdot) : L \to L, \quad z \mapsto A_2(x, z)$$

Stable Operator: Preparation

Proposition

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.
For every pair $(x, y) \in L^2$, the following operators are $\leqslant$-monotone:

$$A_1(\cdot, y) : L \to L, \quad z \mapsto A_1(z, y) \qquad \text{and} \qquad A_2(x, \cdot) : L \to L, \quad z \mapsto A_2(x, z)$$

Proof.

1. Let $x_1 \leqslant x_2$ and $y \in L$.

Stable Operator: Preparation

Proposition

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.
For every pair $(x, y) \in L^2$, the following operators are $\leqslant$-monotone:

$$A_1(\cdot, y) : L \to L, \quad z \mapsto A_1(z, y) \qquad \text{and} \qquad A_2(x, \cdot) : L \to L, \quad z \mapsto A_2(x, z)$$

Proof.

1. Let $x_1 \leqslant x_2$ and $y \in L$.
   Then $(x_1, y) \leq_i (x_2, y)$ and $A(x_1, y) \leq_i A(x_2, y)$, thus $A_1(x_1, y) \leqslant A_1(x_2, y)$.

Stable Operator: Preparation

Proposition

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.
For every pair $(x, y) \in L^2$, the following operators are $\leqslant$-monotone:

$$A_1(\cdot, y) : L \to L, \quad z \mapsto A_1(z, y) \qquad \text{and} \qquad A_2(x, \cdot) : L \to L, \quad z \mapsto A_2(x, z)$$

Proof.

1. Let $x_1 \leqslant x_2$ and $y \in L$.
   Then $(x_1, y) \leq_i (x_2, y)$ and $A(x_1, y) \leq_i A(x_2, y)$, thus $A_1(x_1, y) \leqslant A_1(x_2, y)$.
2. Let $x \in L$ and $y_1 \leqslant y_2$.

Stable Operator: Preparation

Proposition

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.
For every pair $(x, y) \in L^2$, the following operators are $\leqslant$-monotone:

$$A_1(\cdot, y) : L \to L, \quad z \mapsto A_1(z, y) \qquad \text{and} \qquad A_2(x, \cdot) : L \to L, \quad z \mapsto A_2(x, z)$$

Proof.

1. Let $x_1 \leqslant x_2$ and $y \in L$.
   Then $(x_1, y) \leq_i (x_2, y)$ and $A(x_1, y) \leq_i A(x_2, y)$, thus $A_1(x_1, y) \leqslant A_1(x_2, y)$.
2. Let $x \in L$ and $y_1 \leqslant y_2$.
   Then $(x, y_2) \leq_i (x, y_1)$ and $A(x, y_2) \leq_i A(x, y_1)$, thus $A_2(x, y_1) \leqslant A_2(x, y_2)$.

□

- $A_1(\cdot, y)$ has a $\leqslant$-least fixpoint, denoted $\text{lfp}(A_1(\cdot, y))$;
- $A_2(x, \cdot)$ has a $\leqslant$-least fixpoint, denoted $\text{lfp}(A_2(x, \cdot))$.

Stable Operator: Definition

Definition

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.
The **stable approximator** for A is given by $A^{st} : L^2 \to L^2$ with

$$A_1^{st} : L^2 \to L, \qquad (x, y) \mapsto \mathrm{lfp}(A_1(\cdot, y))$$
$$A_2^{st} : L^2 \to L, \qquad (x, y) \mapsto \mathrm{lfp}(A_2(x, \cdot))$$

- A_1^{st}: improve lower bound for all fixpoints of O at or below upper bound;
- A_2^{st}: obtain tightmost new upper bound (eliminate non-minimal fixpoints).

Proposition

Let (x, y) be a postfixpoint of approximator A. Then

$$a \in [\bot, y] \text{ implies } A_1(a, y) \in [\bot, y] \quad \text{and} \quad b \in [x, \top] \text{ implies } A_2(x, b) \in [x, \top].$$

In particular, $\mathrm{lfp}(A_1(\cdot, y)) \leqslant y$ and $x \leqslant \mathrm{lfp}(A_2(x, \cdot))$.

Stable Operator: Observations

Theorem

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.

1. A^{st} is $\leq_i$-monotone.
2. If (x, y) is a consistent postfixpoint of A, then $A^{\text{st}}(x, y)$ is consistent.

Proof.

1. Let $(u, v) \leq_i (x, y)$.

Stable Operator: Observations

Theorem

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.

1. A^{st} is $\leq_i$-monotone.
2. If (x, y) is a consistent postfixpoint of A, then $A^{\text{st}}(x, y)$ is consistent.

Proof.

1. Let $(u, v) \leq_i (x, y)$. Now $y \leqslant v$ implies $A_1(z, v) \leqslant A_1(z, y)$ for all $z \in L$ since A is $\leq_i$-monotone.

Stable Operator: Observations

Theorem

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.

1. A^{st} is $\leq_i$-monotone.
2. If (x, y) is a consistent postfixpoint of A, then $A^{\mathrm{st}}(x, y)$ is consistent.

Proof.

1. Let $(u, v) \leq_i (x, y)$. Now $y \leqslant v$ implies $A_1(z, v) \leqslant A_1(z, y)$ for all $z \in L$ since A is $\leq_i$-monotone. In particular, for $z^* = \mathrm{lfp}(A_1(\cdot, y))$, $A_1(z^*, v) \leqslant A_1(z^*, y) = z^*$ whence z^* is a prefixpoint of $A_1(\cdot, v)$.

Stable Operator: Observations

Theorem

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.

1. A^{st} is $\leq_i$-monotone.
2. If (x, y) is a consistent postfixpoint of A, then $A^{\text{st}}(x, y)$ is consistent.

Proof.

1. Let $(u, v) \leq_i (x, y)$. Now $y \leqslant v$ implies $A_1(z, v) \leqslant A_1(z, y)$ for all $z \in L$ since A is $\leq_i$-monotone. In particular, for $z^* = \text{lfp}(A_1(\cdot, y))$, $A_1(z^*, v) \leqslant A_1(z^*, y) = z^*$ whence z^* is a prefixpoint of $A_1(\cdot, v)$. Thus $\text{lfp}(A_1(\cdot, v)) \leqslant z^* = \text{lfp}(A_1(\cdot, y))$.

Stable Operator: Observations

Theorem

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.

1. A^{st} is $\leq_i$-monotone.
2. If (x, y) is a consistent postfixpoint of A, then $A^{\mathrm{st}}(x, y)$ is consistent.

Proof.

1. Let $(u, v) \leq_i (x, y)$. Now $y \leqslant v$ implies $A_1(z, v) \leqslant A_1(z, y)$ for all $z \in L$ since A is $\leq_i$-monotone. In particular, for $z^* = \mathrm{lfp}(A_1(\cdot, y))$, $A_1(z^*, v) \leqslant A_1(z^*, y) = z^*$ whence z^* is a prefixpoint of $A_1(\cdot, v)$. Thus $\mathrm{lfp}(A_1(\cdot, v)) \leqslant z^* = \mathrm{lfp}(A_1(\cdot, y))$. In combination, $A_1^{\mathrm{st}}(u, v) = \mathrm{lfp}(A_1(\cdot, v)) \leqslant \mathrm{lfp}(A_1(\cdot, y)) = A_1^{\mathrm{st}}(x, y)$.

Stable Operator: Observations

Theorem

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.

1. A^{st} is $\leq_i$-monotone.
2. If (x, y) is a consistent postfixpoint of A, then $A^{\text{st}}(x, y)$ is consistent.

Proof.

1. Let $(u, v) \leq_i (x, y)$. Now $y \leqslant v$ implies $A_1(z, v) \leqslant A_1(z, y)$ for all $z \in L$ since A is $\leq_i$-monotone. In particular, for $z^* = \text{lfp}(A_1(\cdot, y))$, $A_1(z^*, v) \leqslant A_1(z^*, y) = z^*$ whence z^* is a prefixpoint of $A_1(\cdot, v)$. Thus $\text{lfp}(A_1(\cdot, v)) \leqslant z^* = \text{lfp}(A_1(\cdot, y))$. In combination, $A_1^{\text{st}}(u, v) = \text{lfp}(A_1(\cdot, v)) \leqslant \text{lfp}(A_1(\cdot, y)) = A_1^{\text{st}}(x, y)$. A_2^{st}: dual.

Stable Operator: Observations

Theorem

Let $(L, \leqslant)$ be a complete lattice and A be an approximator on $(L^2, \leq_i)$.

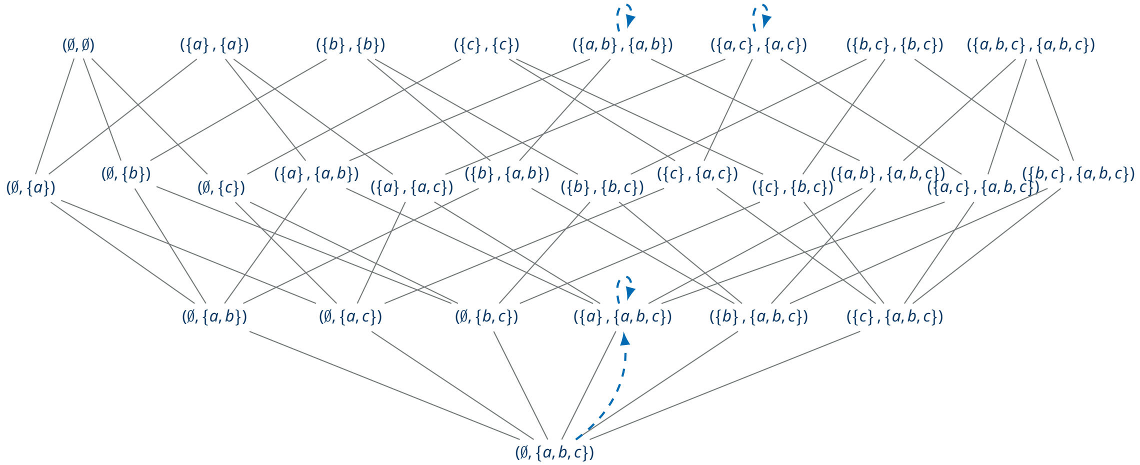
1. A^{st} is $\leq_i$-monotone.
2. If (x, y) is a consistent postfixpoint of A, then $A^{st}(x, y)$ is consistent.

Proof.

1. Let $(u, v) \leq_i (x, y)$. Now $y \leqslant v$ implies $A_1(z, v) \leqslant A_1(z, y)$ for all $z \in L$ since A is $\leq_i$-monotone. In particular, for $z^* = \text{lfp}(A_1(\cdot, y))$, $A_1(z^*, v) \leqslant A_1(z^*, y) = z^*$ whence z^* is a prefixpoint of $A_1(\cdot, v)$. Thus $\text{lfp}(A_1(\cdot, v)) \leqslant z^* = \text{lfp}(A_1(\cdot, y))$. In combination, $A_1^{st}(u, v) = \text{lfp}(A_1(\cdot, v)) \leqslant \text{lfp}(A_1(\cdot, y)) = A_1^{st}(x, y)$. A_2^{st}: dual.
2. Let $x \leqslant y$ with $(x, y) \leq_i A(x, y)$. For every $z \in L$ with $x \leqslant z \leqslant y$, we have $A_1^{st}(x, y) \leqslant A_1^{st}(z, z) = \text{lfp}(A_1(\cdot, z)) \leqslant z \leqslant \text{lfp}(A_2(z, \cdot)) = A_2^{st}(z, z) \leqslant A_2^{st}(x, y)$.

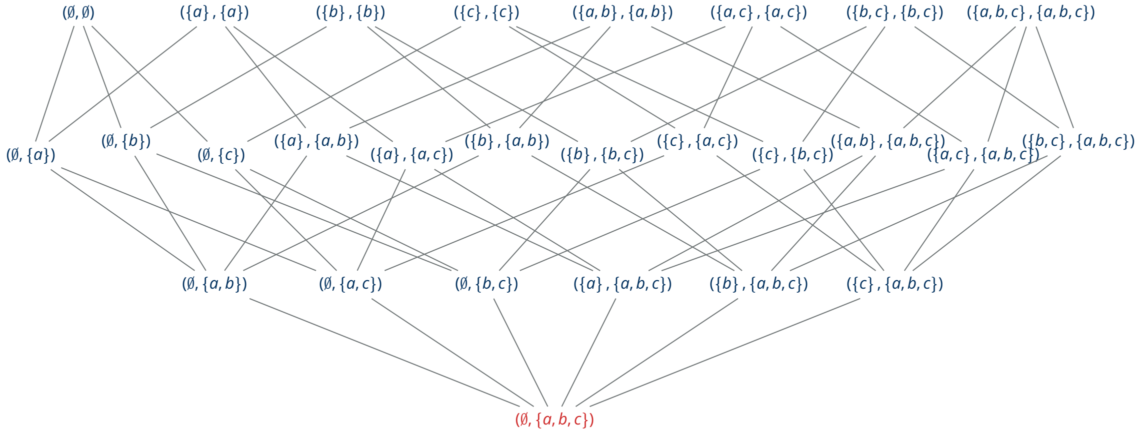
□

Stable Operator ${}_P\mathcal{T}^{\mathrm{st}}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

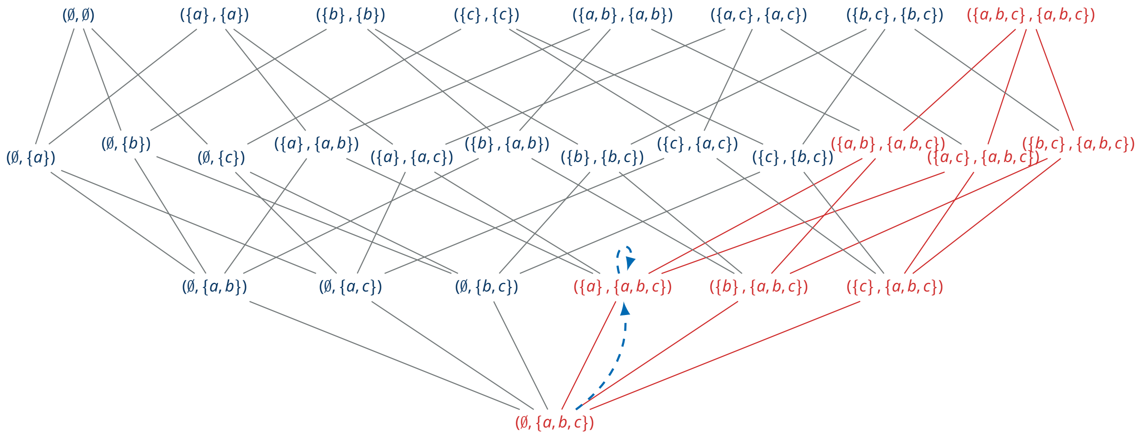
Stable Operator ${}_P\mathcal{T}^{\mathrm{st}}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

${}_P\mathcal{T}^{\mathrm{st}}(\emptyset, \{a, b, c\}) = (\mathrm{lfp}({}_P\mathcal{T}_1(\cdot, \{a, b, c\})), \mathrm{lfp}({}_P\mathcal{T}_2(\emptyset, \cdot)))$

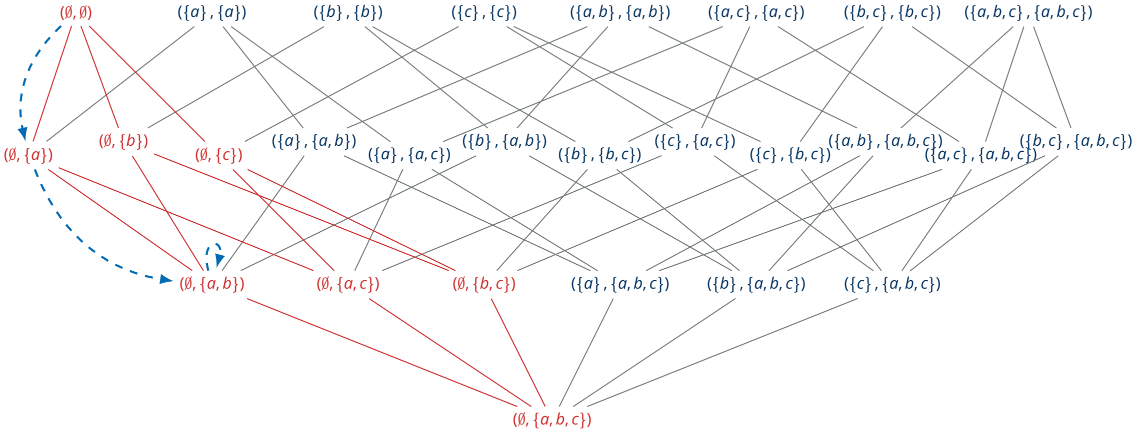
Stable Operator ${}_{P}\mathcal{T}^{\mathrm{st}}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

${}_{P}\mathcal{T}^{\mathrm{st}}(\emptyset, \{a, b, c\}) = (\mathrm{lfp}({}_{P}\mathcal{T}_1(\cdot, \{a, b, c\})), \mathrm{lfp}({}_{P}\mathcal{T}_2(\emptyset, \cdot)))$

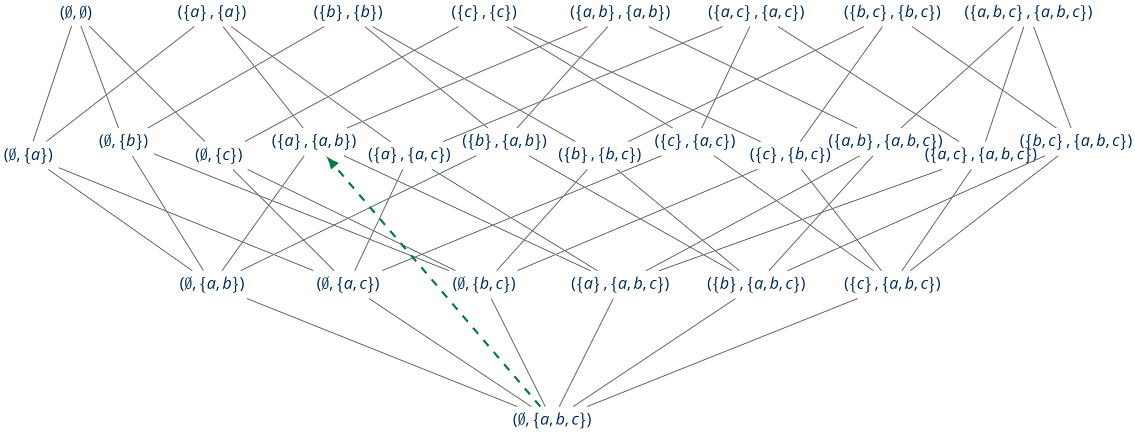
Stable Operator $_{P}\mathcal{T}^{\mathrm{st}}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c\}$

$_{P}\mathcal{T}^{\mathrm{st}}(\emptyset, \{a, b, c\}) = (\{a\}, \mathrm{lfp}(_{P}\mathcal{T}_2(\emptyset, \cdot)))$

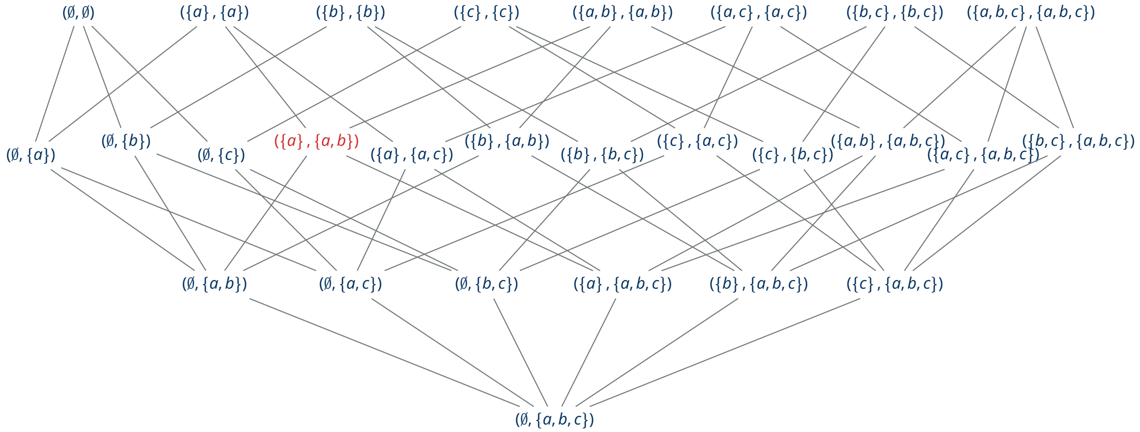
Stable Operator ${}_P\mathcal{T}^{st}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

${}_P\mathcal{T}^{st}(\emptyset, \{a, b, c\}) = (\{a\}, \{a, b\})$

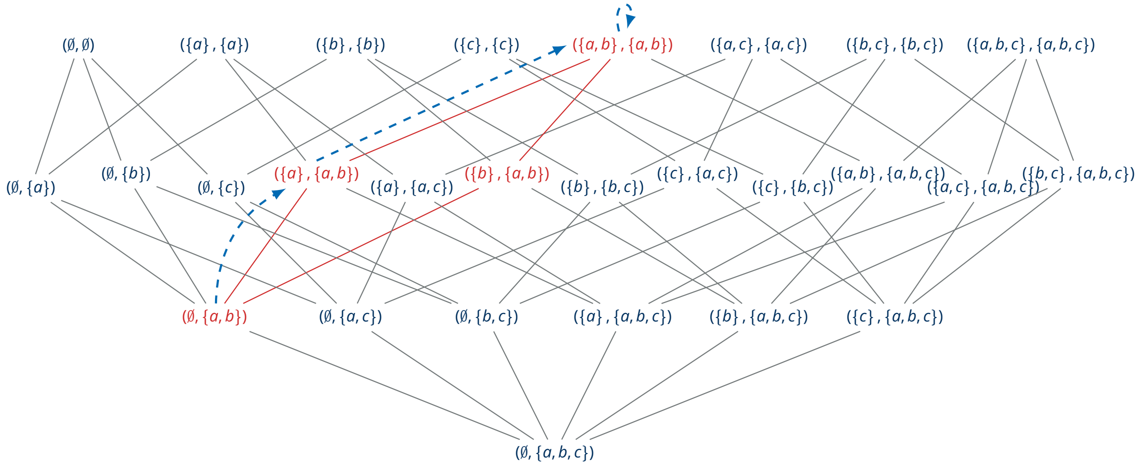
Stable Operator ${}_P\mathcal{T}^{\mathrm{st}}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

${}_P\mathcal{T}^{\mathrm{st}}(\{a\}, \{a, b\}) = (\mathrm{lfp}({}_P\mathcal{T}_1(\cdot, \{a, b\})), \mathrm{lfp}({}_P\mathcal{T}_2(\{a\}, \cdot)))$

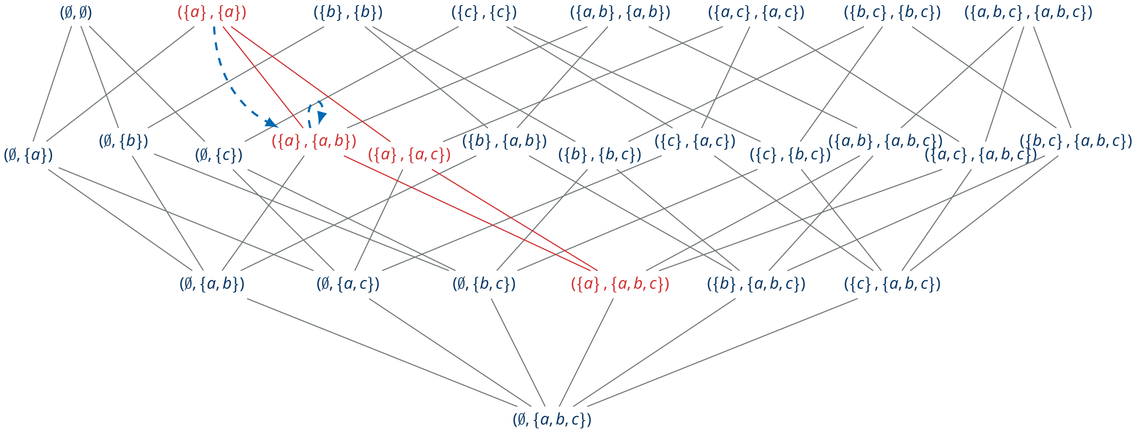
Stable Operator ${}_P\mathcal{T}^{\mathrm{st}}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

${}_P\mathcal{T}^{\mathrm{st}}(\{a\}, \{a, b\}) = (\mathrm{lfp}({}_P\mathcal{T}_1(\cdot, \{a, b\})), \mathrm{lfp}({}_P\mathcal{T}_2(\{a\}, \cdot)))$

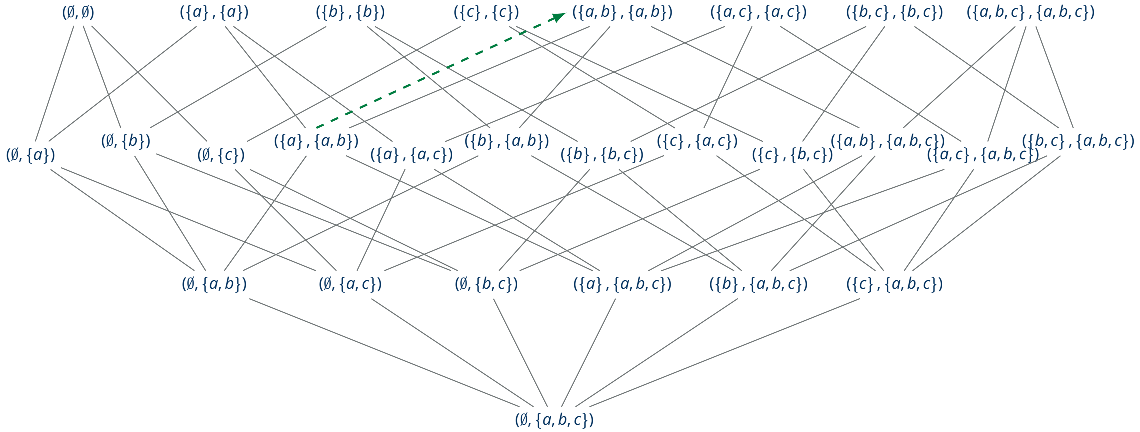
Stable Operator ${}_P\mathcal{T}^{st}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

${}_P\mathcal{T}^{st}(\{a\}, \{a, b\}) = (\{a, b\}, \mathrm{lfp}({}_P\mathcal{T}_2(\{a\}, \cdot)))$

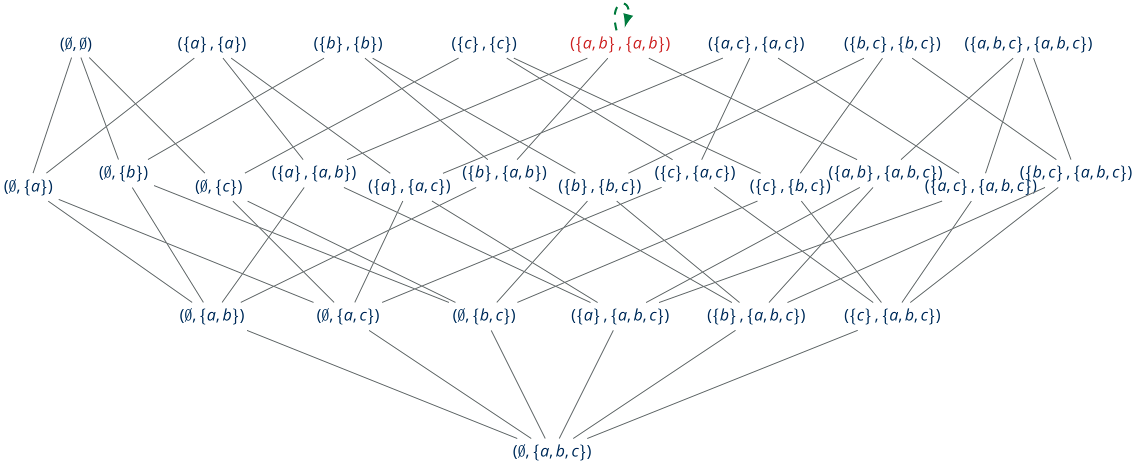
Stable Operator ${}_{P}\mathcal{T}^{\mathrm{st}}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, {\sim}c, \quad c \leftarrow c\}$

${}_{P}\mathcal{T}^{\mathrm{st}}(\{a\}, \{a, b\}) = (\{a, b\}, \{a, b\})$

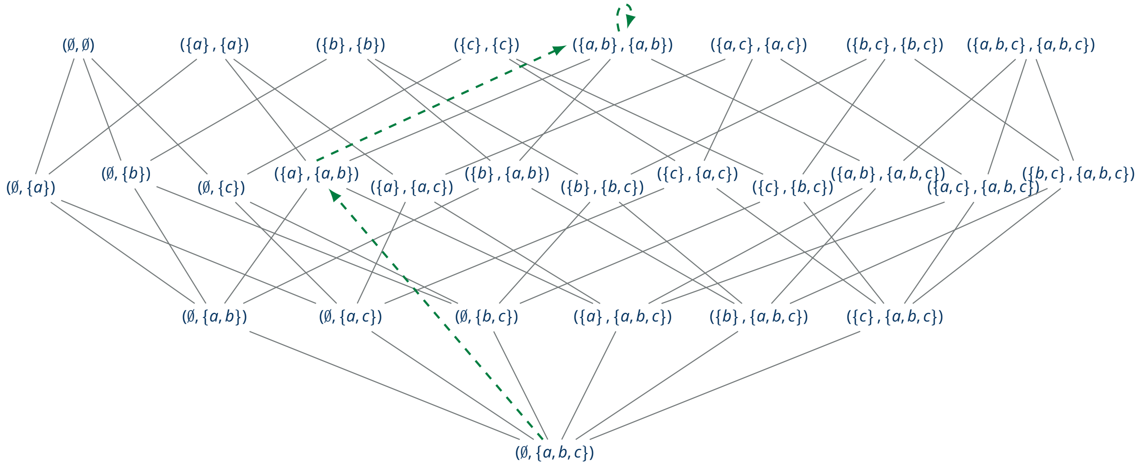
Stable Operator ${}_P\mathcal{T}^{\mathrm{st}}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

${}_P\mathcal{T}^{\mathrm{st}}(\{a, b\}, \{a, b\}) = ({}_PT(\{a, b\}), {}_PT(\{a, b\})) = (\{a, b\}, \{a, b\})$

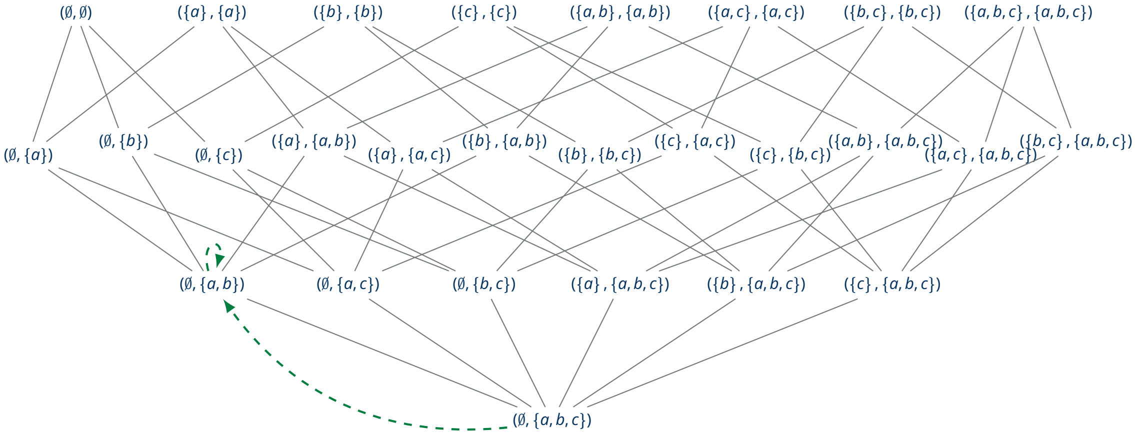
Stable Operator ${}_P\mathcal{J}^{st}$: Example



$P_1 = \{a \leftarrow, \quad b \leftarrow a, \sim c, \quad c \leftarrow c\}$

$\text{lfp}({}_P\mathcal{J}^{st}) = (\{a, b\}, \{a, b\})$: well-founded semantics of P_1

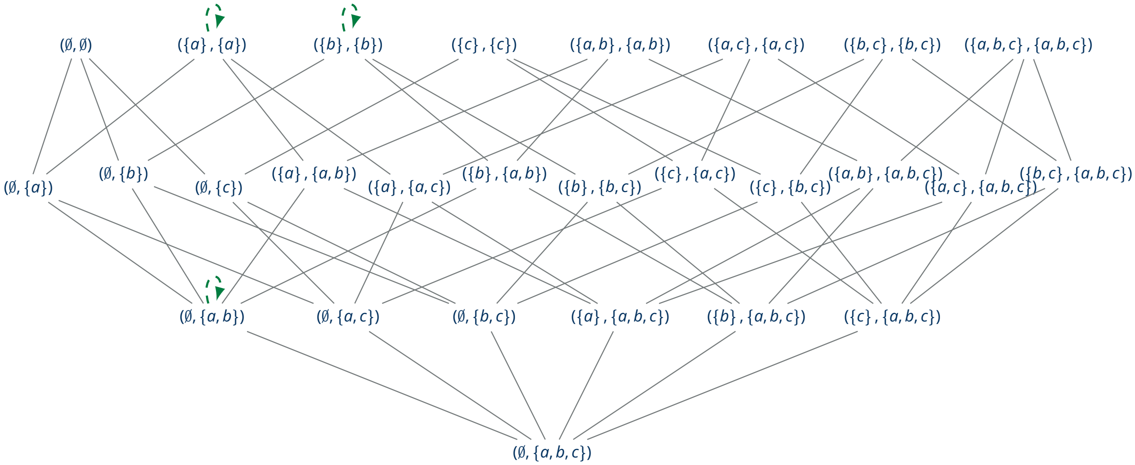
Stable Operator $_{P}\mathcal{J}^{st}$: Example



$P_2 = \{a \leftarrow {\sim}b, \quad b \leftarrow {\sim}a, \quad c \leftarrow c\}$

lfp($_{P}\mathcal{J}^{st}$): well-founded semantics of P_2

Stable Operator $_P\mathcal{T}^{\text{st}}$: Example



$P_2 = \{a \leftarrow \sim b, \quad b \leftarrow \sim a, \quad c \leftarrow c\}$

three-valued stable models of P_2

Stable Semantics: Definition via Operators

Definition

Let $(L, \leqslant)$ be a complete lattice, $O : L \to L$ be an operator.
Let $A : L^2 \to L^2$ be an approximator of O in $(L^2, \leq_i)$. A pair $(x, y) \in L^2$ is

- a **two-valued stable model of** A iff $x = y$ and $A^{\text{st}}(x, y) = (x, y)$;
- a **three-valued stable model of** A iff $x \leqslant y$ and $A^{\text{st}}(x, y) = (x, y)$;
- the **well-founded model of** A iff it is the least fixpoint of A^{st}.

Names inspired by notions from logic programming.

Theorem

1. $\text{lfp}(A) \leq_i \text{lfp}(A^{\text{st}})$;
2. $A^{\text{st}}(x, y) = (x, y)$ implies $A(x, y) = (x, y)$;
3. if $A^{\text{st}}(x, x) = (x, x)$ then x is a $\leqslant$-minimal fixpoint of O;

Reprise: How to Find an Approximator?

Definition

Let $O : L \to L$ be an operator in a complete lattice $(L, \leqslant)$.
Define the **ultimate approximator of** O as follows:

$$U_O : L^2 \to L^2, \qquad (x, y) \mapsto \left(\bigwedge \{O(z) \mid x \leqslant z \leqslant y\}, \bigvee \{O(z) \mid x \leqslant z \leqslant y\}\right)$$

Intuition: Consider glb and lub of applying O pointwise to given interval.

Theorem

For every approximator A of O and consistent pair $(x, y) \in L^2$, we find

$$A(x, y) \leq_i U_O(x, y)$$

Ultimate approximator is most precise approximator possible.

Used e.g. for (PSP-)semantics of aggregates in logic programming.

Conclusion

Conclusion

Summary

- Operators in complete lattices can be used to define semantics of KR formalisms.
- Approximation fixpoint theory provides a general account of operator-based semantics.
- Stable approximator reconstructs well-founded and stable model semantics of logic programming.

Outlook

AFT can be used to show correspondence of ...

- ... extensions of default theories with stable models of logic programs;
- ... expansions of autoepistemic theories with supported models of LPs;
- ... semantics of argumentation frameworks with semantics of LPs.