An algebraic notion of conditional independence, and its application to knowledge representation

Jesse Heyninck

April 25, 2024

Open Universiteit, the Netherlands

Motivating Example

$$r: \ \ \mathtt{inf}(\mathtt{X}) \leftarrow \mathtt{inf}(\mathtt{Y}), \mathtt{cnct}(\mathtt{Y}, \mathtt{X}), \mathrm{not}\ \mathtt{vac}(\mathtt{X}).$$

Motivating Example

$$r_1: \ \texttt{inf}(\texttt{b}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{b}), \mathrm{not}\ \texttt{vac}(\texttt{b}).$$

$$r_2: \ \texttt{inf}(\texttt{c}) \leftarrow \texttt{inf}(\texttt{d}), \texttt{cnct}(\texttt{d}, \texttt{c}), \mathrm{not}\ \texttt{vac}(\texttt{c}).$$

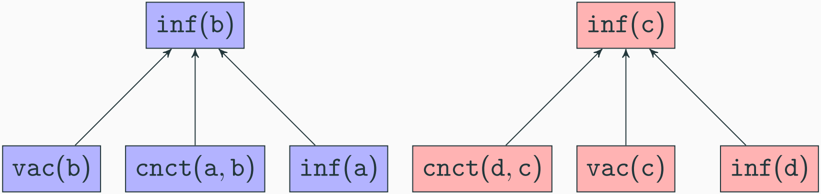
$$r_3: \ \texttt{inf}(\texttt{a}).,\ r_4: \ \texttt{cnct}(\texttt{a}, \texttt{b}).,\ r_5: \ \texttt{cnct}(\texttt{d}, \texttt{c}).$$

Motivating Example

$$r_1 : \ \mathtt{inf(b)} \leftarrow \mathtt{inf(a)}, \mathtt{cnct(a,b)}, \mathrm{not}\ \mathtt{vac(b)}.$$

$$r_2 : \ \mathtt{inf(c)} \leftarrow \mathtt{inf(d)}, \mathtt{cnct(d,c)}, \mathrm{not}\ \mathtt{vac(c)}.$$

$$r_3 : \ \mathtt{inf(a)}., \ r_4 : \ \mathtt{cnct(a,b)}., \ r_5 : \ \mathtt{cnct(d,c)}.$$



Motivating Example

$$r_1: \ \texttt{inf}(\texttt{b}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{b}), \mathrm{not}\ \texttt{vac}(\texttt{b}).$$

$$r_2: \ \texttt{inf}(\texttt{c}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{c}), \mathrm{not}\ \texttt{vac}(\texttt{c}).$$

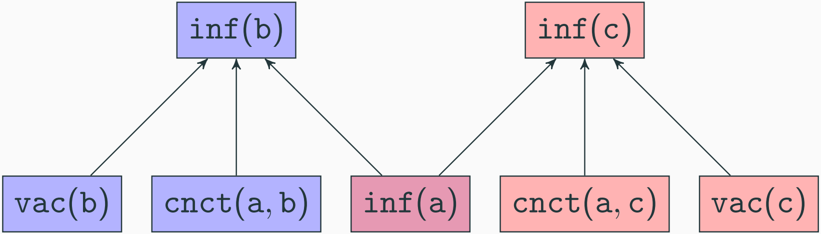
$$r_3: \ \texttt{inf}(\texttt{a}).,\ r_4: \ \texttt{cnct}(\texttt{a}, \texttt{b}).,\ r_5: \ \texttt{cnct}(\texttt{a}, \texttt{c}).$$

Motivating Example

$$r_1: \ \texttt{inf}(\texttt{b}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{b}), \text{not } \texttt{vac}(\texttt{b}).$$

$$r_2: \ \texttt{inf}(\texttt{c}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{c}), \text{not } \texttt{vac}(\texttt{c}).$$

$$r_3: \ \texttt{inf}(\texttt{a}).,\ r_4: \ \texttt{cnct}(\texttt{a}, \texttt{b}).,\ r_5: \ \texttt{cnct}(\texttt{a}, \texttt{c}).$$

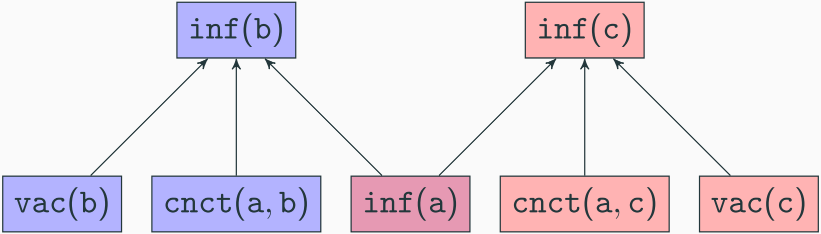


Motivating Example

$$r_1: \ \texttt{inf}(\texttt{b}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{b}), \text{not } \texttt{vac}(\texttt{b}).$$

$$r_2: \ \texttt{inf}(\texttt{c}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{c}), \text{not } \texttt{vac}(\texttt{c}).$$

$$r_3: \ \texttt{inf}(\texttt{a})., \ r_4: \ \texttt{cnct}(\texttt{a}, \texttt{b})., \ r_5: \ \texttt{cnct}(\texttt{a}, \texttt{c}).$$



Once we know `inf(a)` (or $\neg$`inf(a)`), we obtain two independent subprograms.

The NMR-jungle

- Should we investigate conditional independence for normal logic programs? Or logic programs with aggregates, disjunction, choice constructs, ...

The NMR-jungle

- Should we investigate conditional independence for normal logic programs? Or logic programs with aggregates, disjunction, choice constructs, ...
- Under the stable semantics? Or the well-founded, partial stable, supported, regular, ... model semantics?

The NMR-jungle

- Should we investigate conditional independence for normal logic programs? Or logic programs with aggregates, disjunction, choice constructs, ...
- Under the stable semantics? Or the well-founded, partial stable, supported, regular, ... model semantics?
- And what about abstract (dialectical) argumentation (with weights?), autoepistemic logic, default logic, ...?

The NMR-jungle

- Should we investigate conditional independence for normal logic programs? Or logic programs with aggregates, disjunction, choice constructs, ...
- Under the stable semantics? Or the well-founded, partial stable, supported, regular, ... model semantics?
- And what about abstract (dialectical) argumentation (with weights?), autoepistemic logic, default logic, ...?
- Conditional independence seems a good candidate for a top-down approach.

Approximation Fixpoint Theory [DMT00]

- Constructive techniques for approximating the fixpoints of an operator O over a lattice L.
- Uniform framework for the mechanisms underlying many different knowledge representation formalisms, such as logic programming [PDB07], autoepistemic logic [DMT03], default logic [DMT03], abstract argumentation [SW15] and abstract dialectical frameworks [SW15].
- To define semantics for a formalism, the user merely has to choose the lattice and define the operator, and then AFT does all the hard work for the user.

Lattices and operators

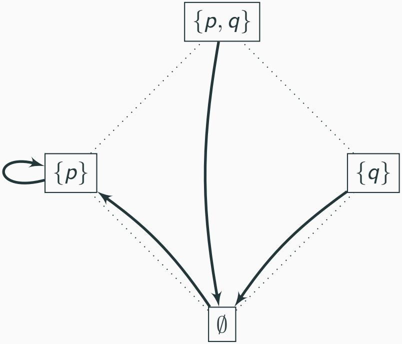
Given a lattice $\mathcal{L} = \langle L, \leq \rangle$, we are interested in operator $O : L \to L$ and its fixpoints.

Lattices and operators

Given a lattice $\mathcal{L} = \langle L, \leq \rangle$, we are interested in operator $O : L \to L$ and its fixpoints.

$\mathcal{P} = \{p \leftarrow \neg q\}$

$\mathrm{IC}_{\mathcal{P}}(x) = \{\alpha \in \mathcal{A}_{\mathcal{P}} \mid \alpha \leftarrow \phi \in \mathcal{P} \text{ and } x(\phi) = \mathsf{T}\}$

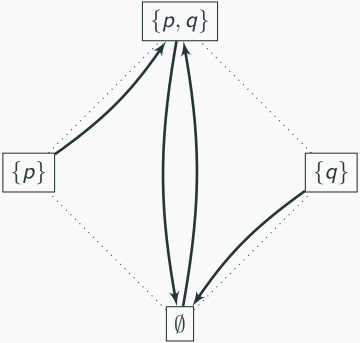


Lattices and operators

Given a lattice $\mathcal{L} = \langle L, \leq \rangle$, we are interested in operator $O : L \to L$ and its fixpoints.

$\mathcal{P} = \{p \leftarrow \neg q; q \leftarrow \neg q\}$

$\mathrm{IC}_{\mathcal{P}}(x) = \{\alpha \in \mathcal{A}_{\mathcal{P}} \mid \alpha \leftarrow \phi \in \mathcal{P} \text{ and } x(\phi) = \mathsf{T}\}$

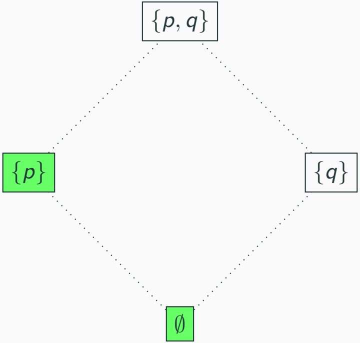


Lattices and operators

Given a lattice $\mathcal{L} = \langle L, \leq \rangle$, we are interested in operator $O : L \to L$ and its fixpoints.

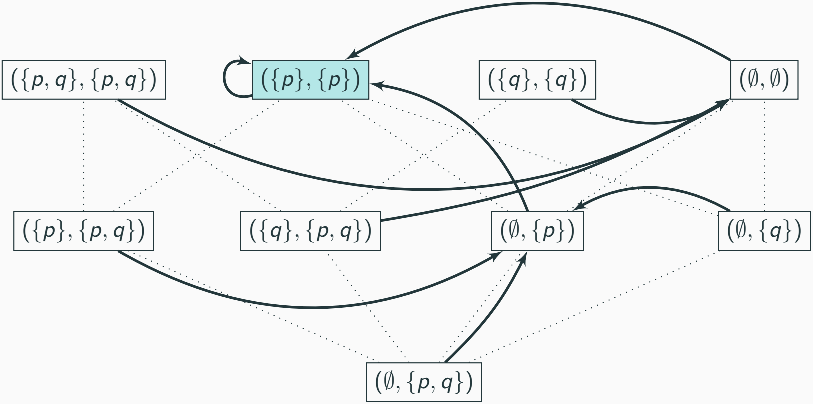
$\mathcal{P} = \{p \leftarrow \neg q; q \leftarrow \neg q\}$

$\mathcal{IC}_{\mathcal{P}}(x, y) = \{\alpha \in \mathcal{A}_{\mathcal{P}} \mid \alpha \leftarrow \phi \in \mathcal{P} \text{ and } (x, y)(\phi) \geq \mathsf{T}\}$



Lattices and operators

Given a lattice $\mathcal{L} = \langle L, \leq \rangle$, we are interested in operator $O : L \to L$ and its fixpoints.



AFT: summary

- Represent or define NMR-formalism by specifying a lattice and a (family of) operator(s).
- Many operators are non-monotonic and obtain a natural semantics in terms of approximations, i.e. pairs of elements.
- These approximations are again operators over lattices.

AFT: summary

- Represent or define NMR-formalism by specifying a lattice and a (family of) operator(s).
- Many operators are non-monotonic and obtain a natural semantics in terms of approximations, i.e. pairs of elements.
- These approximations are again operators over lattices.

Formalism	Lattice	Operator
Logic programming	Herbrand bases ordered by $\subseteq$	Immediate consequence
Formal argumentation	Sets of arguments ordered by $\subseteq$	Defense
ADFs	Sets of arguments ordered by $\subseteq$	Γ-operator
Default logic	Sets of possible worlds ordered by $\supseteq$	Immediate consequence
Weighted ADFs	Multivalued assignments	Γ-operator
...	...	...

AFT: summary

- Represent or define NMR-formalism by specifying a lattice and a (family of) operator(s).
- Many operators are non-monotonic and obtain a natural semantics in terms of approximations, i.e. pairs of elements.
- These approximations are again operators over lattices.

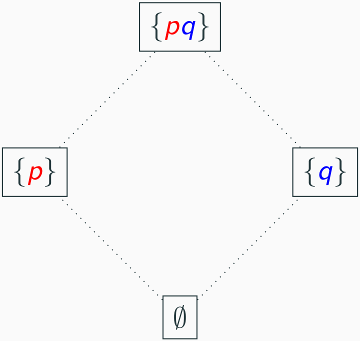
Formalism	Lattice	Operator
Logic programming	Herbrand bases ordered by $\subseteq$	Immediate consequence
Formal argumentation	Sets of arguments ordered by $\subseteq$	Defense
ADFs	Sets of arguments ordered by $\subseteq$	Γ-operator
Default logic	Sets of possible worlds ordered by $\supseteq$	Immediate consequence
Weighted ADFs	Multivalued assignments	Γ-operator
...	...	...

If we study a concept in AFT, it is applicable to many formalisms under a wide family of semantics.

Roadmap

We are interested in defining conditional independence of *parts of a lattice* sanctioned by an operator.

1. Divide lattices in sub-lattices
2. Define and study conditional independence of sub-lattices w.r.t. O.



Sub-lattices

Let I be a set, which we call the *index set*, and for each $i \in I$, let L_i be a set. The product set $\otimes_{i \in I} L_i$ is the following set of functions:

$$\bigotimes_{i \in I} L_i = \{f \mid f : I \to \bigcup_{i \in I} L_i \text{ s.t. } \forall i \in I : \ f(i) \in L_i\}$$

Sub-lattices

Let I be a set, which we call the *index set*, and for each $i \in I$, let L_i be a set. The product set $\otimes_{i \in I} L_i$ is the following set of functions:

$$\bigotimes_{i \in I} L_i = \{f \mid f : I \to \bigcup_{i \in I} L_i \text{ s.t. } \forall i \in I : \; f(i) \in L_i\}$$

The product set $\bigotimes_{i \in I} L_i$ contains all ways of selecting one element of every set L_i.

For a finite set $I = \{1, \ldots, n\}$, the product $\otimes_{i \in I} L_i$ is (isomorphic to) the cartesian product $L_1 \times \ldots \times L_n$.

Example

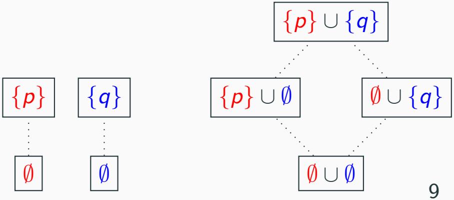
$L_1 = \{\emptyset, \{p\}\}$ *and* $L_2 = \{\emptyset, \{q\}\}$,

$\bigotimes_{i \in \{1,2\}} L_i$ *contains*

$f(1) = f(2) = \emptyset$, *and*

$f'(1) = \emptyset$ *and* $f'(2) = \{q\}$,

$\ldots$



Roadmap

- Motivation and Preliminaries
- Sub-lattices
- Conditional Independence w.r.t. an operator
- Results
- Application to Logic Programming

Conditional Independence w.r.t. an operator

Definition
Let $O : L_1 \otimes L_2 \otimes L_3 \to L_1 \otimes L_2 \otimes L_3$ *be given.*
$L_1 \perp\!\!\!\perp_O L_2 \mid L_3$ *if there exist operators*

$$O_{1,3} : L_1 \otimes L_3 \to L_1 \otimes L_3 \ \text{ and } O_{2,3} : L_2 \otimes L_3 \to L_2 \otimes L_3$$

s.t. for $i, j \in \{1, 2\}$, $i \neq j$, *and for every* $x_i \otimes x_3 \in L_i \otimes L_3$ *and for every* $x_j \in L_j$ *it holds that:*

$$O(x_i \otimes x_j \otimes x_3)_{|i,3} = O_{i,3}(x_i \otimes x_3).$$

Conditional Independence w.r.t. an operator

Definition
Let $O : L_1 \otimes L_2 \otimes L_3 \to L_1 \otimes L_2 \otimes L_3$ *be given.*
$L_1 \perp\!\!\!\perp_O L_2 \mid L_3$ *if there exist operators*

$$O_{1,3} : L_1 \otimes L_3 \to L_1 \otimes L_3 \;\; \text{and} \; O_{2,3} : L_2 \otimes L_3 \to L_2 \otimes L_3$$

s.t. for $i, j \in \{1, 2\}$, $i \neq j$, *and for every* $x_i \otimes x_3 \in L_i \otimes L_3$ *and for every* $x_j \in L_j$ *it holds that:*

$$O(x_i \otimes x_j \otimes x_3)_{|i,3} = O_{i,3}(x_i \otimes x_3).$$

where $x_i \otimes x_j \otimes x_{3|i,3} = x_i \otimes x_3$

Example

$$r_1: \ \texttt{inf}(\texttt{b}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{b}), \text{not } \texttt{vac}(\texttt{b}).$$

$$r_2: \ \texttt{inf}(\texttt{c}) \leftarrow \texttt{inf}(\texttt{a}), \texttt{cnct}(\texttt{a}, \texttt{c}), \text{not } \texttt{vac}(\texttt{c}).$$

$$r_3: \ \texttt{inf}(\texttt{a}).,\ r_4: \ \texttt{cnct}(\texttt{a}, \texttt{b}).,\ r_5: \ \texttt{cnct}(\texttt{a}, \texttt{c}).$$

Example

$$r_1 : \ \mathtt{inf}(\mathtt{b}) \leftarrow \mathtt{inf}(\mathtt{a}), \mathtt{cnct}(\mathtt{a}, \mathtt{b}), \mathrm{not}\ \mathtt{vac}(\mathtt{b}).$$

$$r_2 : \ \mathtt{inf}(\mathtt{c}) \leftarrow \mathtt{inf}(\mathtt{a}), \mathtt{cnct}(\mathtt{a}, \mathtt{c}), \mathrm{not}\ \mathtt{vac}(\mathtt{c}).$$

$$r_3 : \ \mathtt{inf}(\mathtt{a}).,\ r_4 : \ \mathtt{cnct}(\mathtt{a}, \mathtt{b}).,\ r_5 : \ \mathtt{cnct}(\mathtt{a}, \mathtt{c}).$$

$$\begin{aligned} \mathcal{A}_1 &= \{\mathtt{inf}(\mathtt{b}), \mathtt{cnct}(\mathtt{a}, \mathtt{b}), \mathtt{vac}(\mathtt{b})\} \\ \mathcal{A}_2 &= \{\mathtt{inf}(\mathtt{c}), \mathtt{cnct}(\mathtt{a}, \mathtt{c}), \mathtt{vac}(\mathtt{c})\} \\ \mathcal{A}_3 &= \{\mathtt{inf}(\mathtt{a})\} \end{aligned}$$

$\mathcal{P}_1 = \{r_1, r_3, r_4\}$ and $\mathcal{P}_2 = \{r_2, r_3, r_5\}$

Example

$$r_1: \ \mathtt{inf}(\mathtt{b}) \leftarrow \mathtt{inf}(\mathtt{a}), \mathtt{cnct}(\mathtt{a}, \mathtt{b}), \text{not } \mathtt{vac}(\mathtt{b}).$$
$$r_2: \ \mathtt{inf}(\mathtt{c}) \leftarrow \mathtt{inf}(\mathtt{a}), \mathtt{cnct}(\mathtt{a}, \mathtt{c}), \text{not } \mathtt{vac}(\mathtt{c}).$$
$$r_3: \ \mathtt{inf}(\mathtt{a}).,\ r_4: \ \mathtt{cnct}(\mathtt{a}, \mathtt{b}).,\ r_5: \ \mathtt{cnct}(\mathtt{a}, \mathtt{c}).$$

$$\begin{aligned} \mathcal{A}_1 &= \{\mathtt{inf}(\mathtt{b}), \mathtt{cnct}(\mathtt{a}, \mathtt{b}), \mathtt{vac}(\mathtt{b})\} \\ \mathcal{A}_2 &= \{\mathtt{inf}(\mathtt{c}), \mathtt{cnct}(\mathtt{a}, \mathtt{c}), \mathtt{vac}(\mathtt{c})\} \\ \mathcal{A}_3 &= \{\mathtt{inf}(\mathtt{a})\} \end{aligned}$$

$$\mathcal{P}_1 = \{r_1, r_3, r_4\} \text{ and } \mathcal{P}_2 = \{r_2, r_3, r_5\}$$

For any $x_1 \subseteq \mathcal{A}_1$, $x_2 \subseteq \mathcal{A}_2$, $x_3 \subseteq \mathcal{A}_3$,

$$IC_{\mathcal{P}}(x_1 \cup x_2 \cup x_3) \cap (\mathcal{A}_1 \cup \mathcal{A}_3) = IC_{\mathcal{P}_1}(x_1 \cup x_2)$$

Example

$$r_1 : \ \mathtt{inf}(\mathtt{b}) \leftarrow \mathtt{inf}(\mathtt{a}), \mathtt{cnct}(\mathtt{a}, \mathtt{b}), \mathrm{not}\ \mathtt{vac}(\mathtt{b}).$$
$$r_2 : \ \mathtt{inf}(\mathtt{c}) \leftarrow \mathtt{inf}(\mathtt{a}), \mathtt{cnct}(\mathtt{a}, \mathtt{c}), \mathrm{not}\ \mathtt{vac}(\mathtt{c}).$$
$$r_3 : \ \mathtt{inf}(\mathtt{a}).,\ r_4 : \ \mathtt{cnct}(\mathtt{a}, \mathtt{b}).,\ r_5 : \ \mathtt{cnct}(\mathtt{a}, \mathtt{c}).$$

$$\begin{aligned} \mathcal{A}_1 &= \{\mathtt{inf}(\mathtt{b}), \mathtt{cnct}(\mathtt{a}, \mathtt{b}), \mathtt{vac}(\mathtt{b})\} \\ \mathcal{A}_2 &= \{\mathtt{inf}(\mathtt{c}), \mathtt{cnct}(\mathtt{a}, \mathtt{c}), \mathtt{vac}(\mathtt{c})\} \\ \mathcal{A}_3 &= \{\mathtt{inf}(\mathtt{a})\} \end{aligned}$$

$$\mathcal{P}_1 = \{r_1, r_3, r_4\} \text{ and } \mathcal{P}_2 = \{r_2, r_3, r_5\}$$

For any $x_1 \subseteq \mathcal{A}_1$, $x_2 \subseteq \mathcal{A}_2$, $x_3 \subseteq \mathcal{A}_3$,

$$IC_{\mathcal{P}}(x_1 \cup x_2 \cup x_3) \cap (\mathcal{A}_1 \cup \mathcal{A}_3) = IC_{\mathcal{P}_1}(x_1 \cup x_2)$$

(similarly for $\mathcal{P}_2$), and thus:

$$2^{\mathcal{A}_1} \perp\!\!\!\perp_{IC_{\mathcal{P}}} 2^{\mathcal{A}_2} \mid 2^{\mathcal{A}_3}$$

Similarities with Probability Theory

$$A \perp\!\!\!\perp B|C \text{ if } P(A, B|C) = P(A|C)P(B|C).$$

Similarities with Probability Theory

$$A \perp\!\!\!\perp B | C \text{ if } P(A, B|C) = P(A|C)P(B|C).$$

Fact
$L_1 \perp\!\!\!\perp_O L_2 \mid L_3$ *implies:*

$$\begin{aligned} O(x_1 \otimes x_2 \otimes x_3) &= O_{1,3}(x_1 \otimes x_3) \quad \otimes O_{2,3}(x_2 \otimes x_3)_{|2} \\ &= O_{1,2}(x_1 \otimes x_3)_{|1} \otimes O_{2,3}(x_2 \otimes x_3) \end{aligned}$$

Similarities with Probability Theory

$$A \perp\!\!\!\perp B | C \text{ if } P(A, B|C) = P(A|C)P(B|C).$$

Fact

$L_1 \perp\!\!\!\perp_O L_2 \mid L_3$ *implies:*

$$\begin{aligned} O(x_1 \otimes x_2 \otimes x_3) &= O_{1,3}(x_1 \otimes x_3) \quad \otimes O_{2,3}(x_2 \otimes x_3)_{|2} \\ &= O_{1,2}(x_1 \otimes x_3)_{|1} \otimes O_{2,3}(x_2 \otimes x_3) \end{aligned}$$

Furthermore, for any $i, j = 1, 2$, $i \neq j$, $x_i \in \mathcal{L}_i, x_j, x'_j \in \mathcal{L}_j$ *and* $x_3 \in \mathcal{L}_3$ *it holds that:*

$$O(x_i \otimes x_j \otimes x_3)_{|i,3} = O(x_i \otimes x'_j \otimes x_3)_{|i,3}$$

Search for Fixpoints can be Split

Proposition

Let an operator O s.t. $L_1 \perp\!\!\!\perp_O L_2 \mid L_3$ be given. Then $x_1 \otimes x_2 \otimes x_3 = O(x_1 \otimes x_2 \otimes x_3)$ iff $x_1 \otimes x_3 = O_{1,3}(x_1 \otimes x_3)$ and $x_2 \otimes x_3 = O_{2,3}(x_2 \otimes x_3)$.

Search for Fixpoints can be Split

Proposition

Let an operator O s.t. $L_1 \perp\!\!\!\perp_O L_2 \mid L_3$ be given.
Then $x_1 \otimes x_2 \otimes x_3 = O(x_1 \otimes x_2 \otimes x_3)$ iff
$x_1 \otimes x_3 = O_{1,3}(x_1 \otimes x_3)$ and $x_2 \otimes x_3 = O_{2,3}(x_2 \otimes x_3)$.

Example

$r_1 : \ \mathtt{inf(b)} \leftarrow \mathtt{inf(a)}, \mathtt{cnct(a,b)}, \mathrm{not}\ \mathtt{vac(b)}.$

$r_2 : \ \mathtt{inf(c)} \leftarrow \mathtt{inf(a)}, \mathtt{cnct(a,c)}, \mathrm{not}\ \mathtt{vac(c)}.$

$r_3 : \ \mathtt{inf(a)}.,\ r_4 : \ \mathtt{cnct(a,b)}.,\ r_5 : \ \mathtt{cnct(a,c)}.$

We can look for supported models of $\mathcal{P}$ by looking for supported models of $\mathcal{P}_1$ and $\mathcal{P}_2$ and combining them afterwards.

Monotonicity is Preserved

Proposition
Let an operator O s.t. $L_1 \perp\!\!\!\perp_O L_2 \mid L_3$ be given.
Then $O : \bigotimes_{i\in\{1,2,3\}} L_i \to \bigotimes_{i\in\{1,2,3\}} L_i$ is $\leq_\otimes$-monotonic iff $O_{i,3} : L_i \otimes L_3 \to L_i \otimes L_3$ is $\leq_\otimes^{i,3}$-monotonic for $i = 1, 2$.

Proposition
Let a $\leq_\otimes$-monotonic operator O s.t. $L_1 \perp\!\!\!\perp_O L_2 \mid L_3$ be given.
Then x is a least fixed point of O iff $x_{|i,3}$ is a least fixed point of $O_{i,3}$ (for $i = 1, 2$).

Application to AFT I

Proposition
Let $\mathcal{O}$ be an approximation operator s.t. $L_1^2 \perp\!\!\!\perp_{\mathcal{O}} L_2^2 \mid L_3^2$. Then:

- *(x, y) is the Kripke-Kleene fixpoint of $\mathcal{O}$ iff $(x_{|i,3}, y_{|i,3})$ is the Kripke-Kleene fp. of $\mathcal{O}_{i,3}$ for $i = 1, 2$.*
- *(x, y) is a fixpoint of $\mathcal{O}$ iff $(x_{|i,3}, y_{|i,3})$ is a fp. of $\mathcal{O}_{i,3}$ for $i = 1, 2$.*

Proposition
Let $\mathcal{O}$ be an approximation operator s.t. $L_1^2 \perp\!\!\!\perp_{\mathcal{O}} L_2^2 \mid L_3^2$. Then $L_1^2 \perp\!\!\!\perp_{\mathcal{C}(\mathcal{O}_l)} L_2^2 \mid L_3^2$ and $L_1^2 \perp\!\!\!\perp_{\mathcal{C}(\mathcal{O}_u)} L_2^2 \mid L_3^2$.

Application to AFT I

Proposition

Let $\mathcal{O}$ be an approximation operator s.t. $L_1^2 \perp\!\!\!\perp_{\mathcal{O}} L_2^2 \mid L_3^2$. Then:

1. (x, y) *is a fixpoint of* $S(\mathcal{O})$ *iff* $(x_{|i,3}, y_{|i,3})$ *is a fp. of* $S(\mathcal{O}_{i,3})$ *for* $i = 1, 2$.
2. (x, y) *is the well-founded fixpoint of* $\mathcal{O}$ *iff* $(x_{|i,3}, y_{|i,3})$ *is the well-founded fp. of* $\mathcal{O}_{i,3}$ *for* $i = 1, 2$.

Proposition

Let $\mathcal{O}$ be an approximation operator s.t. $L_1^2 \perp\!\!\!\perp_{\mathcal{O}} L_2^2 \mid L_3^2$. Then:

1. *(x, y) is a fixpoint of $S(\mathcal{O})$ iff $(x_{|i,3}, y_{|i,3})$ is a fp. of $S(\mathcal{O}_{i,3})$ for $i = 1, 2$.*
2. *(x, y) is the well-founded fixpoint of $\mathcal{O}$ iff $(x_{|i,3}, y_{|i,3})$ is the well-founded fp. of $\mathcal{O}_{i,3}$ for $i = 1, 2$.*

These results show that most reasoning tasks in NMR can be tackled using a *divide-and-conquer*-methodology if the problem instance admits conditional independencies.

Application to Logic Programming

Proposition
Let a normal logic program $\mathcal{P}$ be given for which $\mathcal{A}_{\mathcal{P}}$ is partitioned into $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ s.t. $\mathcal{A}_1 \perp\!\!\!\perp_{\mathcal{P}} \mathcal{A}_2 \mid \mathcal{A}_3$.

- $(x_1 \cup x_2 \cup x_3, y_1 \cup y_2 \cup y_3)$ *is a supported model of* $\mathcal{P}$ *iff* $(x_i \cup x_3, y_i \cup y_3)$ *is a supported model of* $\mathcal{P}_{|\mathcal{A}_i \cup \mathcal{A}_3}$ *(for* $i = 1, 2$*).*
- $(x_1 \cup x_2 \cup x_3, y_1 \cup y_2 \cup y_3)$ *is a 3-valued stable model of* $\mathcal{P}$ *iff* $(x_i \cup x_3, y_i \cup y_3)$ *is a 3-valued stable model of* $\mathcal{P}_{|\mathcal{A}_i \cup \mathcal{A}_3}$ *(for* $i = 1, 2$*).*
- *The [ultimate] well-founded model of* $\mathcal{P}$ *can be obtained as* $(x_1 \cup x_2 \cup x_3, y_1 \cup y_2 \cup y_3)$*, where* $(x_i \cup x_3, y_i \cup y_3)$ *is the well-founded model of* $\mathcal{P}_{\mathcal{A}_j}$ *(for* $i, j = 1, 2$*,* $i \neq j$*).*

Splitting up Reasoning Tasks

Example

$$r_1: \ \mathtt{inf}(\mathtt{b}) \leftarrow \mathtt{inf}(\mathtt{a}), \mathtt{cnct}(\mathtt{a}, \mathtt{b}), \text{not } \mathtt{vac}(\mathtt{b}).$$

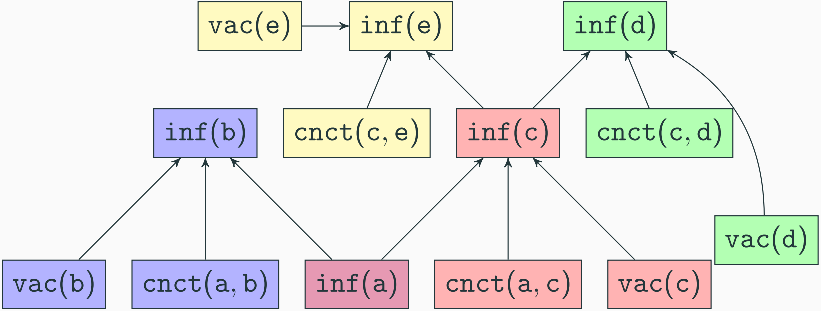
$$r_2: \ \mathtt{inf}(\mathtt{c}) \leftarrow \mathtt{inf}(\mathtt{a}), \mathtt{cnct}(\mathtt{a}, \mathtt{c}), \text{not } \mathtt{vac}(\mathtt{c}).$$

$$r_3: \ \mathtt{inf}(\mathtt{a}).,\ r_4: \ \mathtt{cnct}(\mathtt{a}, \mathtt{b}).,\ r_5: \ \mathtt{cnct}(\mathtt{a}, \mathtt{c}).$$

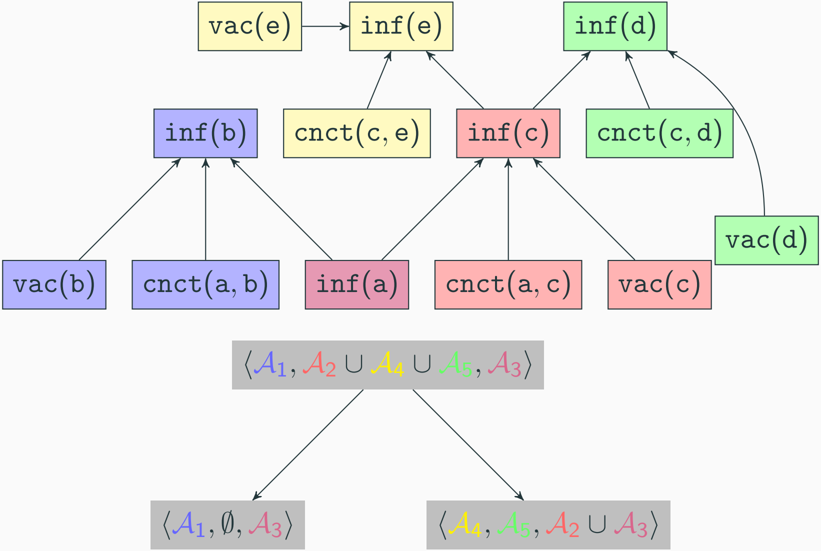
$$r_6: \ \mathtt{inf}(\mathtt{d}) \leftarrow \mathtt{inf}(\mathtt{c}), \mathtt{cnct}(\mathtt{c}, \mathtt{d}), \text{not } \mathtt{vac}(\mathtt{d}).$$

$$r_7: \ \mathtt{inf}(\mathtt{e}) \leftarrow \mathtt{inf}(\mathtt{c}), \mathtt{cnct}(\mathtt{c}, \mathtt{e}), \text{not } \mathtt{vac}(\mathtt{e}).$$

$$r_8: \ \mathtt{cnct}(\mathtt{c}, \mathtt{d}).,\ r_9: \ \mathtt{cnct}(\mathtt{c}, \mathtt{e}).$$



Example


vac(e)
inf(e)
inf(d)
inf(b)
cnct(c, e)
inf(c)
cnct(c, d)
vac(d)
vac(b)
cnct(a, b)
inf(a)
cnct(a, c)
vac(c)
$\langle \mathcal{A}_1, \mathcal{A}_2 \cup \mathcal{A}_4 \cup \mathcal{A}_5, \mathcal{A}_3 \rangle$
$\langle \mathcal{A}_1, \emptyset, \mathcal{A}_3 \rangle$
$\langle \mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_2 \cup \mathcal{A}_3 \rangle$

CIT-Trees

A binary labelled tree (V, E, ν) is a *conditional independence tree* for $\mathcal{O}$ (in short, CIT) if the following holds:

- $\nu : V \to 2^I \times 2^I \times 2^I$,
- the root is labelled $\langle I_1, I_2, I_3 \rangle$ where I_1, I_2, I_3 is a partition of I,
- for every $(v_1, v_2), (v_1, v_3) \in E$, where $\nu(v_i) = \langle I_1^i, I_2^i, I_3^i \rangle$ for $i = 1, 2, 3$,
$I_j^1 \cup I_3^1 = I_1^j \cup I_j^2 \cup I_j^3$ for $j = 2, 3$ and
$I_j^1 \cup I_2^1 \cup I_3^1 = I_1^2 \cup I_2^2 \cup I_3^2 \cup I_1^3 \cup I_2^3 \cup I_3^3$,
- if $\nu(v) = \langle I_1, I_2, I_3 \rangle$ then $\bigotimes_{i \in I_1} \mathcal{L}_i \perp\!\!\!\perp_{\mathcal{O}_{I_1 \cup I_2 \cup I_3}} \bigotimes_{i \in I_2} \mathcal{L}_i \mid \bigotimes_{i \in I_3} \mathcal{L}_i$.

$$\begin{tikzcd}
 & \langle \mathcal{A}_1, \mathcal{A}_2 \cup \mathcal{A}_4 \cup \mathcal{A}_5, \mathcal{A}_3 \rangle \arrow[dl] \arrow[dr] & \\
\langle \mathcal{A}_1, \emptyset, \mathcal{A}_3 \rangle & & \langle \mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_2 \cup \mathcal{A}_3 \rangle
\end{tikzcd}$$

Example: Well-Founded Interpretation

$$\begin{tikzcd}
 & \langle \mathcal{A}_1, \mathcal{A}_2 \cup \mathcal{A}_4 \cup \mathcal{A}_5, \mathcal{A}_3 \rangle \arrow[dl] \arrow[dr] & \\
\langle \mathcal{A}_1, \emptyset, \mathcal{A}_3 \rangle & & \langle \mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_2 \cup \mathcal{A}_3 \rangle
\end{tikzcd}$$

- $\mathrm{WF}(\mathcal{P}_1 \cup \mathcal{P}_3) = \{\mathtt{inf}(\mathrm{a}), \mathtt{cnct}(\mathrm{a}, \mathrm{b}), \mathtt{inf}(\mathrm{b})\}$
  Search space size: 2^4
- $\mathrm{WF}(\mathcal{P}_4 \cup \mathcal{P}_2 \cup \mathcal{P}_3) = \{\mathtt{inf}(\mathrm{a}), \mathtt{cnct}(\mathrm{a}, \mathrm{c}), \mathtt{inf}(\mathrm{c}), \mathtt{cnct}(\mathrm{c}, \mathrm{e}), \mathtt{inf}(\mathrm{e})\}$
  Search space size: 2^7
- $\mathrm{WF}(\mathcal{P}_5 \cup \mathcal{P}_2 \cup \mathcal{P}_3) = \{\mathtt{inf}(\mathrm{a}), \mathtt{cnct}(\mathrm{a}, \mathrm{c}), \mathtt{inf}(\mathrm{c}), \mathtt{cnct}(\mathrm{c}, \mathrm{d}), \mathtt{inf}(\mathrm{d})\}$
  Search space size: 2^7

Example: Well-Founded Interpretation

$$
\begin{tikzcd}[column sep=small]
 & \langle \mathcal{A}_1, \mathcal{A}_2 \cup \mathcal{A}_4 \cup \mathcal{A}_5, \mathcal{A}_3 \rangle \arrow[dl] \arrow[dr] & \\
\langle \mathcal{A}_1, \emptyset, \mathcal{A}_3 \rangle & & \langle \mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_2 \cup \mathcal{A}_3 \rangle
\end{tikzcd}
$$

- $\mathrm{WF}(\mathcal{P}_1 \cup \mathcal{P}_3) = \{\mathtt{inf}(\mathrm{a}), \mathtt{cnct}(\mathrm{a}, \mathrm{b}), \mathtt{inf}(\mathrm{b})\}$
  Search space size: 2^4
- $\mathrm{WF}(\mathcal{P}_4 \cup \mathcal{P}_2 \cup \mathcal{P}_3) = \{\mathtt{inf}(\mathrm{a}), \mathtt{cnct}(\mathrm{a}, \mathrm{c}), \mathtt{inf}(\mathrm{c}), \mathtt{cnct}(\mathrm{c}, \mathrm{e}), \mathtt{inf}(\mathrm{e})\}$
  Search space size: 2^7
- $\mathrm{WF}(\mathcal{P}_5 \cup \mathcal{P}_2 \cup \mathcal{P}_3) = \{\mathtt{inf}(\mathrm{a}), \mathtt{cnct}(\mathrm{a}, \mathrm{c}), \mathtt{inf}(\mathrm{c}), \mathtt{cnct}(\mathrm{c}, \mathrm{d}), \mathtt{inf}(\mathrm{d})\}$
  Search space size: 2^7
- $\mathrm{WF}(\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \cup \mathcal{P}_4 \cup \mathcal{P}_5) =$
  $\mathrm{WF}(\mathcal{P}_1 \cup \mathcal{P}_3) \cup \mathrm{WF}(\mathcal{P}_4 \cup \mathcal{P}_2 \cup \mathcal{P}_3) \cup \mathrm{WF}(\mathcal{P}_5 \cup \mathcal{P}_2 \cup \mathcal{P}_3)$.
  Original search space size: 2^{14}

Fixed-Parameter Complexity

Definition
Let an operator O over the power set lattice $\bigotimes_{i \in I} \mathcal{L}_i$ and CIT $T = (V, E, \nu)$ be given s.t. V_{l} are the leafs of T. The CIT-partition-size of O relative to (V, E, ν) is defined as

$$\max(\{|\bigotimes_{i \in I_j} \mathcal{L}_i \otimes \bigotimes_{i \in I_3} \mathcal{L}_i| \mid v \in V_{\text{l}}, \nu(v) = \langle I_1, I_2, I_3 \rangle, j = 1, 2\})$$

Fixed-Parameter Complexity

Definition
Let an operator O over the power set lattice $\bigotimes_{i \in I} \mathcal{L}_i$ and CIT $T = (V, E, \nu)$ be given s.t. V_l are the leafs of T. The CIT-partition-size of O relative to (V, E, ν) is defined as

$$\max(\{|\bigotimes_{i \in I_j} \mathcal{L}_i \otimes \bigotimes_{i \in I_3} \mathcal{L}_i| \mid v \in V_\text{l}, \nu(v) = \langle I_1, I_2, I_3 \rangle, j = 1, 2\})$$

Proposition
Let a $\leq_\otimes$-monotonic operator O over the product lattice $\bigotimes_{i \in I} \mathcal{L}_i$ and CIT $T = (V, E, \nu)$ with CIT-partition-size s be given. Assume that $O(x, y)$ can be computed to a call to an NP*-oracle. The least fixpoint of O can be computed in time $O(f(s))$.*

Rounding Up

Other things in the paper

- Syntactical (sufficient) criteria for conditional independence in logic programs.
- Comparison with treewidth-based decompositions of logic programs.
- (Non)-satisfaction of graphoid properties.
- Comparison with Darwiche's logical notion of independence [Dar97].
- Relations with splitting in AFT [VGD06].

Conclusion

- Algebraic account of conditional independence of sublattice w.r.t. an operator.
- Language-independent account of how the structure of problems in fixpoint-based logic allows to reduce global to parallel instances of local problems.
- Immediately applicable to logic program, formal argumentation, ADFs, default logic, . . .

Conclusion

- Algebraic account of conditional independence of sublattice w.r.t. an operator.
- Language-independent account of how the structure of problems in fixpoint-based logic allows to reduce global to parallel instances of local problems.
- Immediately applicable to logic program, formal argumentation, ADFs, default logic, ...
- Future work:
  - Implementation (happy to join forces)

Conclusion

- Algebraic account of conditional independence of sublattice w.r.t. an operator.
- Language-independent account of how the structure of problems in fixpoint-based logic allows to reduce global to parallel instances of local problems.
- Immediately applicable to logic program, formal argumentation, ADFs, default logic, . . .
- Future work:
  - Implementation (happy to join forces)
  - Application to your favorite formalism?

Conclusion

- Algebraic account of conditional independence of sublattice w.r.t. an operator.
- Language-independent account of how the structure of problems in fixpoint-based logic allows to reduce global to parallel instances of local problems.
- Immediately applicable to logic program, formal argumentation, ADFs, default logic, ...
- Future work:
  - Implementation (happy to join forces)
  - Application to your favorite formalism?

Thank you for your attention. Questions?

Adnan Darwiche.
A logical notion of conditional independence: properties and applications.
Artificial Intelligence, 97(1-2):45–82, 1997.

Marc Denecker, Victor Marek, and Mirosław Truszczyński.
Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning.
In *Logic-based Artificial Intelligence*, volume 597 of *The Springer International Series in Engineering and Computer Science*, pages 127–144. Springer, 2000.

Marc Denecker, Victor Marek, and Mirosław Truszczyński.
Uniform semantic treatment of default and autoepistemic logics.
Artificial Intelligence, 143(1):79–122, 2003.

Nikolay Pelov, Marc Denecker, and Maurice Bruynooghe.
Well-founded and stable semantics of logic programs with aggregates.
Theory and Practice of Logic Programming, 7(3):301–353, 2007.

Bibliography iii

Hannes Strass and Johannes Peter Wallner.
Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory.
Artificial Intelligence, 226:34–74, 2015.

Joost Vennekens, David Gilis, and Marc Denecker.
Splitting an operator: Algebraic modularity results for logics with fixpoint semantics.
ACM Transactions on computational logic (TOCL), 7(4):765–797, 2006.