We don’t have a clue how the mind is working.

Non-monotonic reasoning — domain-theoretic perspectives

Pascal Hitzler

Artificial Intelligence Institute, Dresden University, Germany

Contents

- Non-monotonic reasoning.
- Formal concept analysis.
- Domain theory.

- Non-monotonic reasoning with conceptual knowledge.

What is non-monotonic reasoning?

Inspired by commonsense reasoning.

…acting under incomplete knowledge …

…jumping to conclusions …

Tweety is a bird, hence flies.
But you may find out later that it is a penguin …

Seek abstract (high-level) knowledge representation and reasoning formalisms suitable for this kind of reasoning.

Different formalisms for NMR . . .

- Axiomatic aproaches (Gabbay 1985; Kraus, Lehmann, Magidor 1990)

  supraclassicality: $\frac{X \vdash \alpha}{X \mid\!\sim \alpha}$

  cautious monotonicity: $\frac{X \mid\!\sim \alpha \quad X \mid\!\sim \beta}{X, \alpha \mid\!\sim \beta}$ $\qquad \left(\text{monotonicity: } \frac{X \vdash \alpha}{X, \beta \vdash \alpha}\right)$

- Modality for *belief* (Moore's Autoepistemic Logic 1984)

- Second-order approaches (McCarthy's Circumscription 1977)

Default Logic (Reiter 1980)

(propositional case, F, G, H formulae)

default: $\frac{F:G}{H}$ "if F, and if G is possible, then H"

Δ set of defaults. E is called a *default extension* of Δ if E is a minimal logically closed theory (set of formulae) satisfying:
whenever $E \models F$ and G is consistent with E, then $H \in E$.

$$\frac{\text{bird} : \neg\text{penguin}}{\text{flies}}$$

NMR with logic programs

Horn program: set of CNF formulae (clauses) $p \vee \neg q_1 \vee \cdots \vee \neg q_n$

written: $p \leftarrow q_1, \ldots, q_n$

Set of *atoms* inferred depends monotonically on program.

procedurally (Prolog): $p \leftarrow r$ infers $\neg p$

after addition of $r \leftarrow$ we infer p

non-monotonic behaviour of negation

closed world assumption

negation as “(finite) failure to prove it”

Stable models

next step: allow negation in clauses: $p \leftarrow q_1, \ldots, q_n, \neg r_1, \ldots, \neg r_m$.
(*normal* logic program)

Intended semantics approach: what *should* it be?
(deviating from Prolog)

Interpret each clause as *default*

$$\frac{q_1 \wedge \cdots \wedge q_n : \neg(r_1 \vee \cdots \vee r_m)}{p}.$$

Default extensions of a program are exactly the (logical closures of the) *stable models* of a program (Gelfond & Lifschitz 1991).

$$\text{flies} \leftarrow \text{bird}, \neg\text{penguin}$$

Stable models: fixed point characterization

Horn program P, I set of atoms (*interpretation*):

$T_P(I) = \{p \mid \exists(p \leftarrow p_1, \ldots, p_n) \in P. \forall i. p_i \in I\}$.

T_P Scott-continuous, monotonic, least fixed point

$\mathrm{fix}(T_P) = \bigcup T_P^n(\emptyset)$.

$\mathrm{fix}(T_P)$ is *least model* of P.

normal program P: Set P/I to be the Horn program consisting of all

$p \leftarrow p_1, \ldots, p_n$ generated from all

$p \leftarrow p_1, \ldots, p_n, \neg q_1, \ldots, \neg q_m$ with $\forall i. q_i \notin I$.

Stable models characterized by: $I = \mathrm{fix}(T_{P/I}) \quad (= \mathrm{GL}_P(I))$.

GL_P antitonic (not monotonic in general).

$$\text{flies} \vee \text{penguin} \leftarrow \text{bird}$$

Answer sets

Syntactic extension: $p_1 \vee \cdots \vee p_k \leftarrow q_1 \wedge \cdots \wedge q_m \wedge \neg r_1 \wedge \cdots \wedge \neg r_n$

written: $p_1, \ldots, p_k \leftarrow q_1, \ldots, q_m, \neg r_1, \ldots, \neg r_n.$

I interpretation (set of atoms), P program. P/I defined as before, resulting in program with rules of the form $p_1, \ldots, p_k \leftarrow q_1, \ldots, q_m.$

These have minimal models $\mathrm{Min}(P/I)$.

I *answer set* if $I \in \mathrm{Min}(P/I)$.

NMR systems: dlv (Vienna), smodels (Helsinki)

Formal Concept Analysis (FCA)

(FCA: an approach to data mining and analysis; Ganter & Wille 1999)

G set of objects; M set of attributes. $C \subseteq G \times M$ *formal context.*

$A \subseteq G$ then $A' = \{m \in M \mid (\forall g \in A)(g, m) \in C\}$.
$B \subseteq M$ then $B' = \{g \in G \mid (\forall m \in B)(g, m) \in C\}$.

Formal concept: Pair (A, B) with $A' = B$, $A = B'$.
Equivalently: All (B', B'') for $B \subseteq M$.

Formal concept lattice:
complete lattice of all concepts ordered by $\supseteq$ in second argument.

Formal Concept Analysis

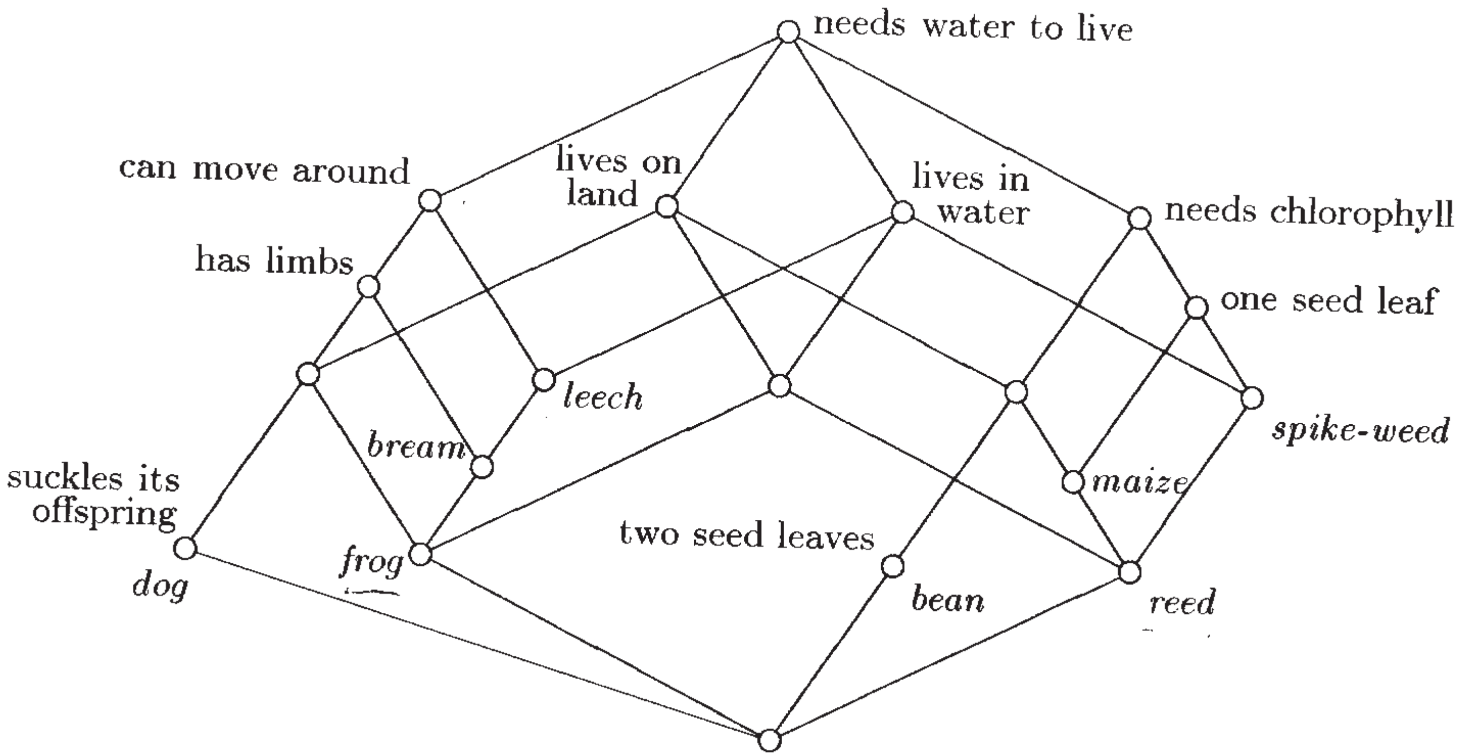


Figure 1.4 Concept lattice for the educational film "Living beings and water".

(source: Ganter & Wille, Formal Concept Analysis, Springer, 1999. (also next slide))

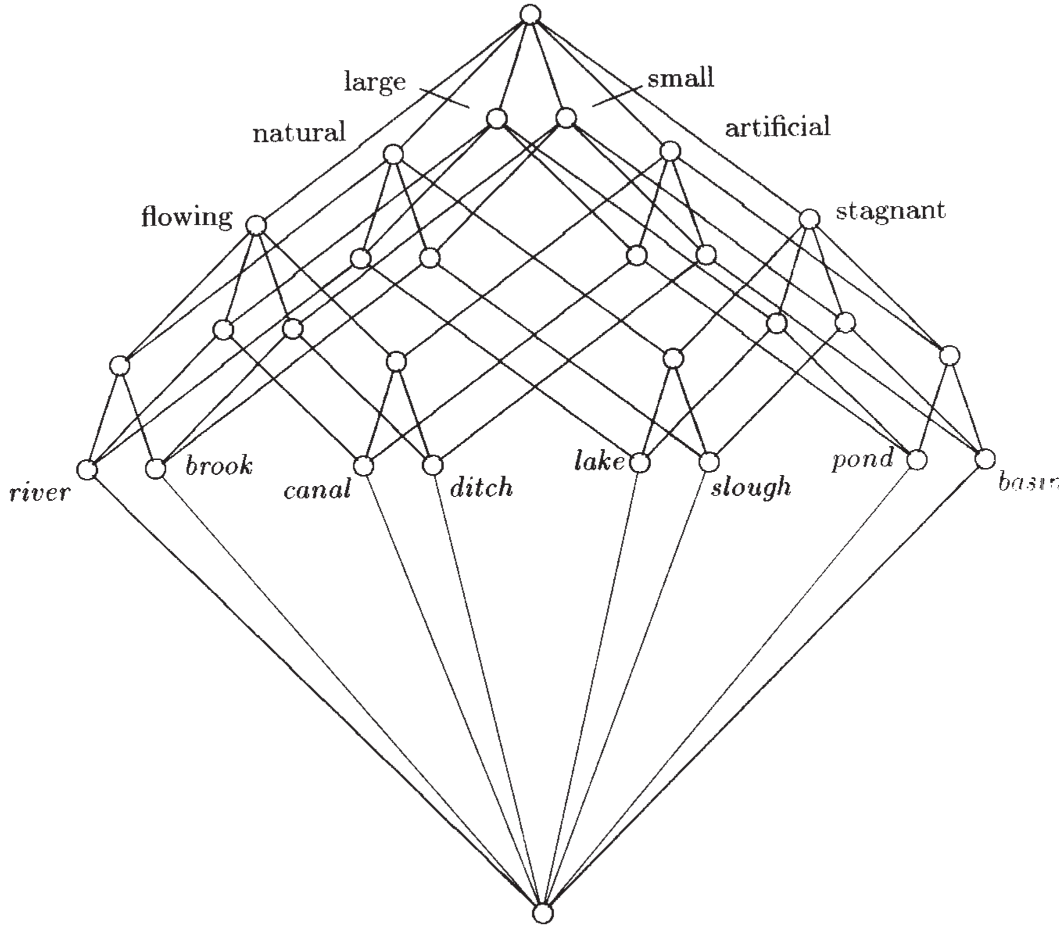


Figure 2.9 An additive line diagram of the concept lattice of a *lexical field "waters"*. The set representation is based on the irreducible attributes, i.e. the positioning of the attribute concepts determines that of all remaining concepts. If we interpret the line segments between the unit element and the attribute concepts as vectors, we obtain the position of an arbitrary concept by the sum of the vectors belonging to attributes of its concept intent starting from the unit element. Other diagrams for the same lattice can be found in Figure 2.10.

Domain theory: coherent algebraic cpos

cpo: directed complete partial order with bottom $(D, \sqsubseteq)$

$c \in \mathsf{K}(D)$ (compact) iff $(\forall A \text{ directed})(d \sqsubseteq \bigsqcup A \implies (\exists a \in A) d \sqsubseteq a)$

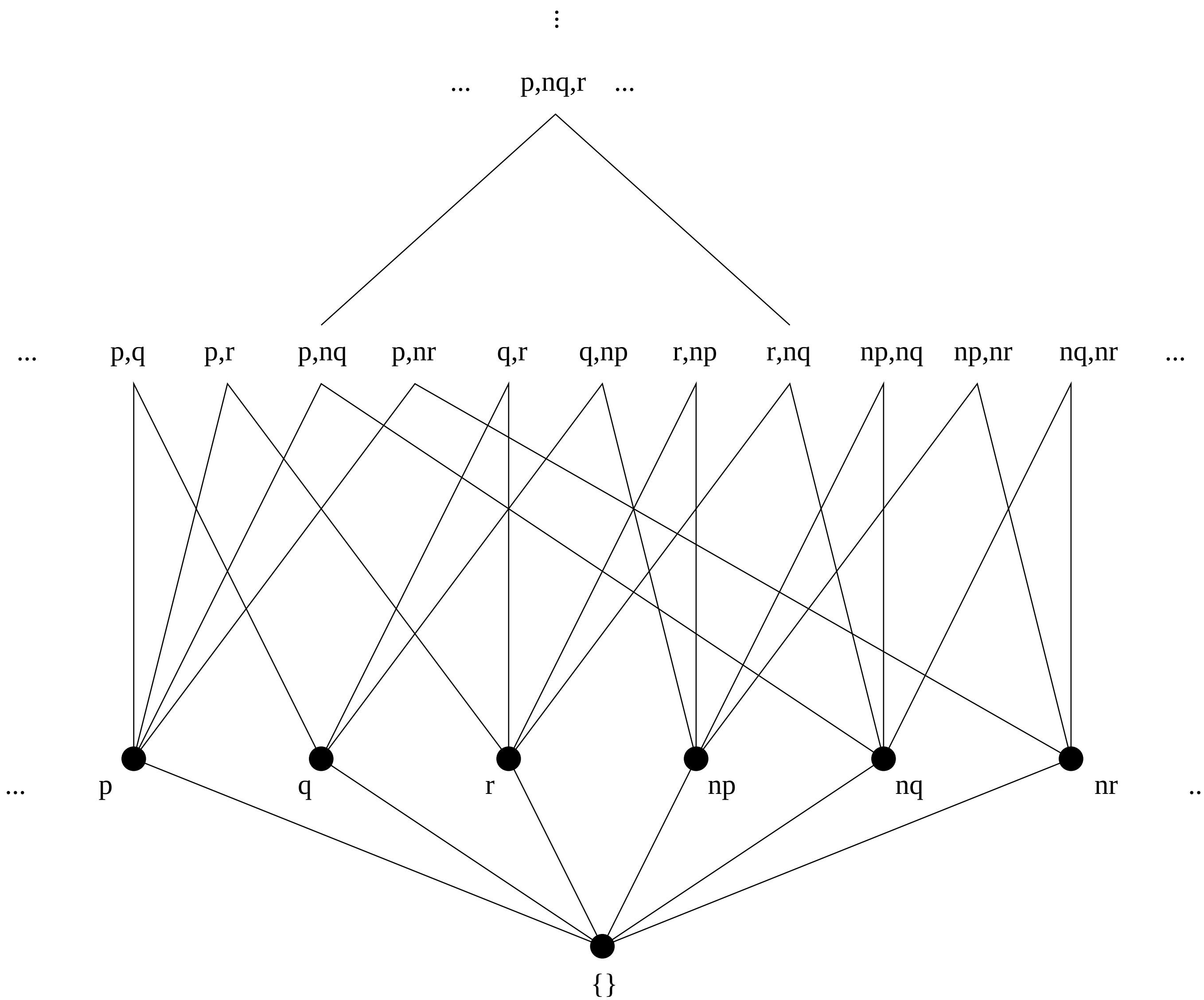
cpo *algebraic*: $(\forall x)\,(x = \bigsqcup (x\!\downarrow \cap \mathsf{K}(D)))$

Scott topology: base $\{\uparrow c \mid c \in \mathsf{K}(D)\}$

coherent: finite intersections of compact-opens are compact-open

Examples: Finite posets with bottom. Powersets. $\mathbb{T}^\omega$.

Plotkin's $\mathbb{T}^\omega$



Logic RZ

(Rounds & Zhang, 2001)

clause X: finite subset of $\mathsf{K}(D)$

$w \in D$: $w \models X$ iff $(\exists x \in X)(x \sqsubseteq w)$.

theory T: set of clauses.

$w \models T$ iff $(\forall X \in T)(w \models X)$.

$T \models X$ iff $(\forall w \in D)(w \models T \implies w \models X)$.

Logic RZ

Proof theory: (WLP'02)

$$\frac{}{\{\bot\}}$$

$$\frac{X; \qquad a \in X; \qquad y \sqsubseteq a}{\{y\} \cup (X \setminus \{a\})}$$

$$\frac{X; \qquad y \in \mathsf{K}(D)}{\{y\} \cup X}$$

$$\frac{X_1 \quad X_2; \qquad a_1 \in X_1 \quad a_2 \in X_2}{\mathrm{mub}\{a_1, a_2\} \cup (X_1 \setminus \{a_1\}) \cup (X_2 \setminus \{a_2\})}$$

Logic RZ is compact.

Smyth powerdomain via the logic RZ

(Rounds & Zhang 2001)

The logically closed theories are the ideals under $\sqsubseteq^\sharp$.

Smyth powerdomain: consistent closed theories under set-inclusion.

Logic RZ and Formal Concept Analysis (FCA)

(with Matthias Wendt, ICCS 2003)

Consider subposet D of all $(\{b\}', \{b\}'')$, $b \in M$,
and all $(\{a\}'', \{a\}')$, $a \in G$, ordered reversely (add $\perp$).

If D is finite, then

for $D \supseteq \{b_i \mid i \in I\} = B \subseteq M$ we have

$$B'' = \{b \in M \mid \{\{b_i\} \mid i \in I\} \models \{b\}\}.$$

Logic RZ and FCA

(Hitzler 2003)

L subposet of all $(\{b\}', \{b\}'')$, $b \in M$, and all $(\{a\}'', \{a\}')$, $a \in G$, ordered reversely.

D coherent algebraic cpo,
$\iota : L \to D$ order-reversing injection which covers all of $\mathsf{K}(D)$.

$B = \{m_1, \ldots, m_n\} \subseteq M$ such that $\iota(m_i) \in \mathsf{K}(D)$ for all i.

Then

$$B'' = \{m \mid \{\{\iota(m_1)\}, \ldots, \{\iota(m_n)\}\} \models \{\iota(m)\}\}.$$

Logic programming in coherent algebraic domains

(Rounds & Zhang 2001)

Add material implication: $X \leftarrow Y$ for X, Y clauses.

$w \models P$: if $w \models Y$ for $X \leftarrow Y \in P$, then $w \models X$.

Propagation rule CP(P):

$$\frac{X_1 \quad \ldots \quad X_n; \quad a_i \in X_i; \quad Y \leftarrow Z \in P; \quad \mathrm{mub}\{a_1, \ldots, a_n\} \models Z}{Y \cup \bigcup_{i=1}^{n} (X_i \setminus \{a_i\})}$$

Semantic operator on theories:

$$\mathcal{T}_P(T) = \mathrm{cons}\left(\{Y \mid Y \text{ is a } \mathrm{CP}(P)\text{-consequence of } T\}\right).$$

- ▶ $\mathcal{T}_P$ is Scott continuous [RZ01].
- ▶ $\mathrm{fix}\,(\mathcal{T}_P) = \mathrm{cons}(P)$.

Additon of default negation

Extended rules: $X \leftarrow Y, {\sim}Z$.

P program, T theory. Define P/T:

Replace $Y, {\sim}Z$ by Y if $T \not\models Z$.

Remove rule if $T \models Z$.

T *answer theory* for P if $T = \text{cons}(P/T) = \text{fix}(\mathcal{T}_{P/T})$.

A version of default logic

Consider $\mathbb{T}^\omega$. (Hitzler 2003)

Clauses are the propositional formulae in disjunctive normal form.
Extended rules correspond to defaults.
Answer theories correspond to default extensions.
But logical consequence is not classical.

On restricted syntax (normal LPs) the following are equivalent:

- answer theories
- default theories
- answer sets (stable models)

On $\mathbb{T}^\omega$ we obtain something akin to propositional default logic.

(It is probably related to work by Rainer Osswald (Hagen), 2003.)

Answer set programming

We do the same with *models*.

P program, $w \in D$. Define P/w:

Replace $Y, \sim Z$ by Y if $w \not\models Z$.

Remove rule if $w \models Z$.

w *min-answer model* for P if w is minimal with $w \models \mathrm{fix}(\mathcal{T}_{P/w})$.

Answer set programming

Consider $\mathbb{T}^\omega$.

Consider programs P with rules $X \leftarrow Y, \sim Z$ such that:

X contains only atoms in $\mathbb{T}^\omega$.

Y is a singleton clause.

Z contains only atoms in $\mathbb{T}^\omega$ or $\perp$.

These programs are exactly extended disjunctive programs.

Min-answer models w correspond to *answer sets* $\{L \text{ atom} \mid w \models \{L\}\}$ and vice-versa.

Conclusions/Further Work

Domain-theoretic perspective of answer set programming.

(cf. also Rounds & Zhang 199x/200x)

- Decidability aspects?
- Concise meta-theory?

Paradigm for reasoning with conceptual knowledge.

(cf. also Zhang 2003, Zhang & Shen 2003)

- Symbolic data analysis applications?