Introduction to lattice theory and generality orderings

A lattice is a system of elements with 2 basic operations: formation of meet and formation of join

Basics of lattice theory

1. Sets
2. Relations and operations
3. Equivalence relations
4. Partial orders
5. Lattices
6. Quasi orders
7. Generality orderings

Relevance to ILP

ILP is concerned with the automatic construction of "general" logical statements from "specific" ones.

– For example, given $mem(1, [1, 2]) \leftarrow$ construct $mem(A, [A|B]) \leftarrow$

Questions:

1. What do the words "general" and "specific" mean in a logical setting?

2. Can statements of increasing (decreasing) generality be enumerated in an orderly manner?

These are questions about the mathematics of "generality"

- ILP identifies "generality" with $\models$. That is, C_1 is "more general" than C_2 iff $C_1 \models C_2$

- The relation $\models$ results in a quasi-ordering over a set of clauses.

- ILP systems are programs that search such quasi-ordered sets

Sets

Fundamental concept in mathematics

- A set S contains *elements* ($S = \{a, b, \ldots\}$); elements are *members* of a set ($a \in S$).

- Two sets S, T are equal ($S = T$) iff they contain precisely the same elements, otherwise $S \neq T$.

- A set T is a subset of S ($T \subseteq S$) if every member of T is a member of S. If $T \subseteq S$ and $S \subseteq T$ then $S = T$. $T \subseteq S$ means $S \supseteq T$.

- If $T \subseteq S$ and S contains an element not in T then T is a proper subset of S ($T \subset S$) $T \subset S$ means $S \supset T$.

Sets (contd.)

Intersection of two sets S, T

- The set with elements in common to sets S and T, denoted by $S \cap T$ or ST or $S \cdot T$. $ST \subseteq S$ and $ST \subseteq T$ for all S, T.

- If S and T are disjoint, ST is denoted by the unique set having no members ($\emptyset$). $\emptyset \subseteq S$ for all S and $\emptyset \cdot S = \emptyset$ for all S.

Union of two sets S, T

- The set with elements which belong to at least S or T, denoted by $S \cup T$ or $S + T$. $S \subseteq S + T$ and $T \subseteq S + T$ for all S, T. $S + \emptyset = S$ for all S.

Equivalence of two sets S, T

- If there is a $1-1$ correspondence between members of S and members of T (every member of S corresponds to just one member of T and every member of T corresponds to just one member of T) then $S \sim T$.

- If there is a $T \subset S$ and $S \sim T$ then S is said to be infinite, otherwise S is said to be finite. The set of natural numbers $\mathcal{N}$ is of particular importance. $\mathcal{N}$ is infinite, and any set $S \sim \mathcal{N}$ is said to be countable.

Relations and Operations

Finite sequence: a set of n elements placed in a 1−1 correspondence to the set $\{1, \ldots, n\}$ arranged in order of succession. An *ordered pair* is a sequence of 2 elements.

Dyadic or binary relation R over a set S: a set of ordered pairs (x, y) where $x, y \in S$. If $(a, b) \in R$ then aRb means "a is in the relation R to b" or "relation R holds between a and b."

Finitary operation in a set S: let $s_n = (x_1, \ldots, x_n)$ be sequences of n elements. To each such sequence, associate just one element $y \in S$. The set P of ordered pairs (s_n, y) is a finitary operation in S. $s_n P y$ denotes a n-ary operation and is denoted by $P(x_1, \ldots, x_n) = y$. If $n = 1$, then P is a dyadic relation over S.

- Let $S = \mathcal{N}$. Then addition $(+)$, subtraction $(-)$ etc. are examples of binary operations in S.

If a n-ary operation P is defined for every sequence s_n of n elements of A, then S is *closed* wrt P. A set S closed wrt one or more finitary operations is called an *algebra*. A *subalgebra* is a subset of an algebra S which is self-contained wrt to the operations.

- $\mathcal{N}$ is closed wrt the binary operations of $+$ and $\times$, and $\mathcal{N}$ along with $+, \times$ form an algebra.

- The set $\mathcal{E}$ of even numbers is a subalgebra of algebra of $\mathcal{N}$ with $+, \times$. The set $\mathcal{O}$ of odd numbers is not a subalgebra.

– Let $S \subseteq U$ and $S' \subseteq U$ be the set with elements of U not in S (the unary operation of complementation). Let $U = \{a, b, c, d\}$. The subsets of U with the operations of complementation, intersection and union form an algebra. How many subalgebras are there of this algebra?

Answer=

Equivalence Relations

Equivalence relation E over a set S is a dyadic relation over S that satisfies the following properties:

Reflexive. For every $a \in S$, aEa

Symmetric. If aEb then bEa

Transitive. If aEb and bEc then aEc

– Let $S = \mathcal{N}$ and aEb iff $a + b$ is even. That is, E consists of all ordered pairs (a, b) whose sum is even. This makes all even numbers equivalent, and the odd numbers equivalent

– Let $S = \{a, b, c, d\}$ and xEy if $x, y \in \{a, b\}$ or $x, y \in \{c, d\}$. This makes a, b equivalent to each other and c, d equivalent to each other

Theorem. Any equivalence relation E over a non-empty set S results in a partition of S into disjoint non-empty subsets which contain all the members of S.

– The subsets are called "equivalence classes" or "blocks of the partition". Some special partitions: every block contains exactly 1 element (zero partition); at most 1 block contains more than 1 element (singular partition); and 1 block contains all the elements (unity partition).

Theorem. Any partition of a set S into disjoint subsets such that every member of S is in some subset results in an equivalence relation E over S.

Partial Order

Gven an equality relation $=$ over elements of a set S, a partial order $\preceq$ over S is a dyadic relation over S that satisfies the following properties:

Reflexive. For every $a \in S$, $a \preceq a$

Anti-Symmetric. If $a \preceq b$ and $b \preceq a$ then $a = b$

Transitive. If $a \preceq b$ and $b \preceq c$ then $a \preceq c$

– If $a \preceq b$ and $a \neq b$ then $a \prec b$

– $b \succeq a$ means $a \preceq b$, $b \succ a$ means $a \prec b$

– If $a \preceq b$ or $b \preceq a$ then a, b are comparable, otherwise they are not comparable

– A set S over which a relation of partial order is defined is called a *partially ordered set*

– It is sometimes convenient to refer to a set S and a relation R defined over S together by the pair $< S, R >$

– Examples of partially ordered sets $< S, \preceq >$:

  * S is a set of sets, $S_1 \preceq S_2$ means $S_1 \subseteq S_2$

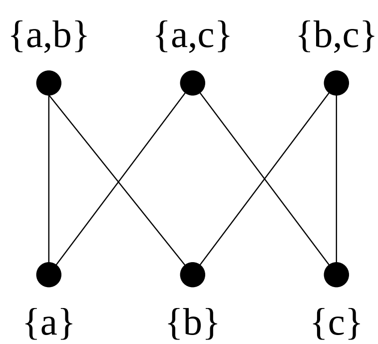
  * $S = \mathcal{N}$, $n_1 \preceq n_2$ means $n_1 = n_2$ or there is a $n_3 \in \mathcal{N}$ such that $n_1 + n_3 = n_2$

  * S is the set of equivalence relations $E_1, \ldots$ over some set T, $E_L \preceq E_M$ means for $u, v \in T$, uE_Lv means uE_Mv (that is, $(u, v) \in E_L$ means $(u, v) \in E_M$).

- Given a set $S = \{a, b, \ldots\}$ if $a \prec b$ and there is no $x \in S$ such that $a \prec x \prec b$ then b *covers* a or a is a *downward cover* of b

- Given a set S let S_{down} be a set of downward covers of $b \in S$. If for all $x \in S$, $x \prec b$ implies there is an $a \in S_{down}$ s.t. $x \preceq a \prec b$, then S_{down} is said to be a *complete* set of downward covers of b.

Diagrammatic representation of a partially ordered set



Partial order (contd.)

Let $< S, \preceq >$ be a p.o. set and $T \subseteq S$

Least element of T $a \in T \; s.t. \; \forall t \in T \; a \preceq t$	Greatest element of T $a \in T \; s.t. \; \forall t \in T \; a \succeq t$
Least element, if it exists, is unique. If $T = S$ this is the "zero" element	Greatest element, if it exists If $T = S$ then this is the "unity" element
Minimal element of T $a \in T \;\; \nexists t \in T \; s.t. \; t \prec a$ Minimal element need not be unique	Maximal element of T $a \in T \;\; \nexists t \in T \; s.t. \; t \succ a$ Maximal element need not be unique
Lower bound of T $b \in S \; s.t. \; b \preceq t \; \forall t \in T$	Upper bound of T $b \in S \; s.t. \; b \succeq t \; \forall t \in T$
Glb g of T $b \preceq g \; \forall b, g : \; lbs \; of \; T$ If it exists, glb is unique	Lub g of T $b \succeq g \; \forall b, g : \; ubs \; of \; T$ If it exists lub is unique

If for every pair $a, b \in S$, $a \prec b$ or $b \prec a$ then S is *totally ordered* or is a *chain*. Any subset of a chain is a chain.

Lattice

A lattice is a partially ordered set $< S, \preceq >$ in which every pair $a, b \in S$ has a greatest lower bound ($a \sqcap b$ or ab or *meet*) in S and a least upper bound ($a \sqcup b$ or $a + b$ or *join*) in S

Theorem. A lattice is an algebra with the binary operations of $\sqcap$ and $\sqcup$

Properties of $\sqcap$ and $\sqcup$

– $a \sqcap b = b \sqcap a$, and $a \sqcup b = b \sqcup a$

– $a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c$, and $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$

– If $a \preceq b$ then $a \sqcap b = a$, and $a \sqcup b = b$

– $a \sqcap (a \sqcup b) = a$, and $a \sqcup (a \sqcap b) = a$

Example

- Let S be all the subsets of $\{a, b, c\}$, and for $X, Y \in S$, $X \preceq Y$ mean $X \subseteq Y$, $X \sqcap Y = X \cap Y$ and $X \sqcup Y = X \cup Y$. Then $< S, \subseteq >$ is a lattice.

Quasi-order

A quasi-order Q in a set S is a dyadic relation over S that satisfies the following properties:

Reflexive. For every $a \in S$, aQa

Transitive. If aQb and bQc then aQc

– Differs from equivalence relation in that symmetry is not required

– Differs from partial order in that no equality is defined, therefore anti-symmetry property cannot be defined

Theorem. If a quasi-order Q is defined on a set $S = \{a, b, \ldots\}$, and we define a dyadic relation E as follows: aEb iff aQb and bQa, then E is an equivalence relation.

Theorem. Let E partition S into subsets $X, Y, \ldots$ of equivalent elements. Let $T = \{X, Y, \ldots\}$ and $\preceq$ be a dyadic relation in T meaning $X \preceq Y$ in T iff xQy in S for some $x \in X, y \in Y$. Then T is partially ordered by $\preceq$.

A quasi-order order Q over a set S results in a partial ordering over a set of equivalence classes of elements in S

In ILP, we will be concerned with cases where S consists logical sentences (atoms and clauses) and Q is the *subsumption* relation or the *implication* relation

Subsumption ordering over atoms

Consider the set S of all atoms in some language, and $S^+ = S \cup \{\top, \bot\}$. Let the dyadic relation $\succeq$ be such that:

- $\top \succeq l$ for all $l \in S^+$
- $l \succeq \bot$ for all $l \in S^+$
- $l \succeq m$ iff there is a substitution θ s.t. $l\theta = m$, for $l, m \in S$

$\succeq$ is a quasi-ordering known as "subsumption". A partial ordering results from the partition of S^+ into the sets $\{[\top]\}, \{[\bot]\}, X_1, \ldots$ where $[l]$ denotes all atoms that are alphabetic variants of l. That is, if $l, m \in X_i$ then there are substitutions μ and σ s.t. $l\mu = m$ and $m\sigma = l$. Thus, $\succeq$ is a partial ordering over the set of equivalence classes of atoms (S_E^+)

Example of subsumption ordering on atoms

- $l = mem(A, [A, B]) \succeq mem(1, [1, 2]) = m$ since with $\theta = \{A/1, B/2\}$, $l\theta = m$

- $mem(A1, [A1, B1]), mem(A2, [A2, B2]) \ldots$ are all members of the same equivalence class

For atoms $l, m \in S$, subsumption is equivalent to implication

- If $l \models m$ then $l \succeq m$

Subsumption lattice of atoms

The p.o. set of equivalence classes of atoms S_E^+ is a lattice with the binary operations $\sqcap$ and $\sqcup$ defined on elements of S_E^+ as follows:

– $[\bot] \sqcap [l] = [\bot]$, and $[\top] \sqcap [l] = [l]$

– If $l_1, l_2 \in S$ have *mgu* θ then $[l_1] \sqcap [l_2] = [l_1\theta] = [l_2\theta]$ otherwise $[l_1] \sqcap [l_2] = [\bot]$

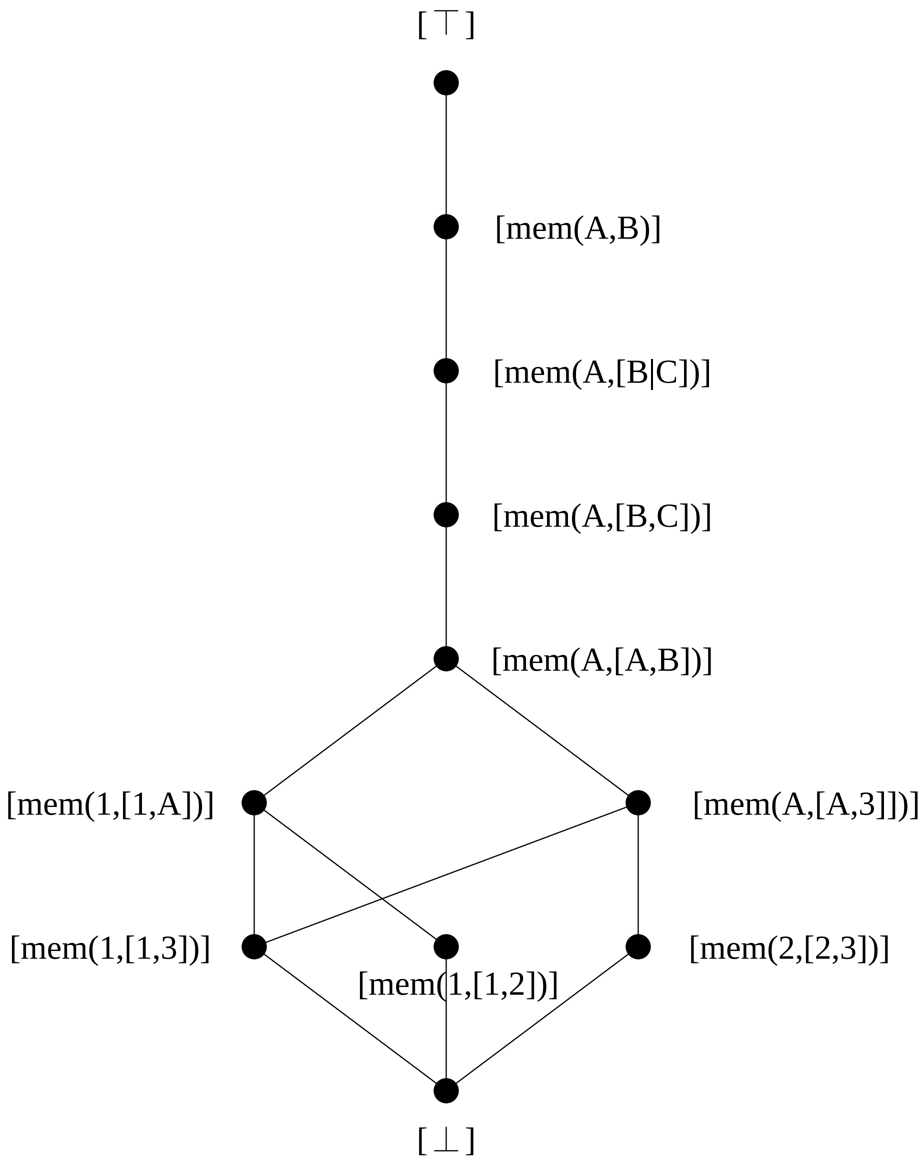
– $[\bot] \sqcup [l] = [l]$, and $[\top] \sqcup l] = [\top]$

– If l_1 and l_2 have *lgg* m then $[l_1] \sqcup [l_2] = [m]$ otherwise $[l_1] \sqcup [l_2] = [\top]$

The join operation or lub called *lgg* stands for least-general-generalisation of atoms (Lab Nos. 5, 6)

Example

$S^{+} = \{ \top, \bot, mem(1,[1,3]), mem(1,[1,2]), mem(2,[2,3]),$
$mem(1,[1,A]), mem(A,[A,B]), mem(A,[A,3]), mem(A,[B,C]),$
$mem(A,[B|C])\ mem(A,B) \}$



Finite Chains in the Lattice

It can be shown that if $l \succ m$ (l covers m) then there is a finite sequence $l_1, \ldots, l_n$ s.t. $l \succ l_1 \succ \ldots l_n$ where l_n is an alphabetic variant of m

Progress from l_i to l_{i+1} is achieved by applying one of the following substitutions:

1. $\{X/f(X_1, \ldots, X_k)\}$ where X is a variable in l_i, $X_1, \ldots, X_k$ are distinct variables that do not appear in l_i, and f is some k-ary function symbol in the language

2. $\{X/c\}$ where X is a variable in l_i, and c is some constant in the language

3. $\{X/Y\}$ where X, Y are distinct variables in l_i

In ILP, these 3 operations define a "downward refinement operator"

Subsumption ordering over Horn clauses

Consider the set S of all Horn clauses in some language, and $S^+ = S \cup \{\bot\}$. Let $\Box$ denote the empty clause and the dyadic relation $\succeq$ be such that:

- $\top = \Box \succeq C$ for all $C \in S^+$
- $C \succeq \bot$ for all $C \in S^+$
- $C \succeq D$ iff there is a substitution θ s.t. $C\theta \subseteq D$, for $C, D \in S$

$\succeq$ is a quasi-ordering known as "subsumption". A partial ordering results from the partition of S^+ into the sets $\{[\bot]\}, X_1, \ldots$ where $[C]$ denotes all clauses that are subsume-equivalent to C. This are not simply alphabetic variants (as in the case of atoms).

That is, if $C, D \in X_i$ there are substitutions μ and σ s.t. $C\mu \subseteq D$ and $D\sigma \subseteq C$. In fact, the subsume-equivalent class of C is infinite, and $[C]$ is usually represented by its "smallest" member (*reduced form*). Thus, $\succeq$ is a partial ordering over the set of subsume-equivalent classes of clauses (S_E^+)

Example of subsumption ordering on clauses

- $C = p(X,Y) \leftarrow \succeq p(a,b) \leftarrow q(a,b) = D$ since with $\theta = \{X/a, Y/b\}$, $C\theta \subseteq D$

- $p(X,X) \leftarrow, p(X,X1) \leftarrow, p(X1,X2) \leftarrow \ldots$ are all in the same equivalence class. $p(X,X) \leftarrow$ is the reduced form of this class.

For clauses $C, D \in S$, subsumption is *not* equivalent to implication

- If $C \succeq D$ then $C \models D$

Subsumption lattice of Horn clauses

The p.o. set of equivalence classes of Horn clauses S_E^+ is a lattice with the binary operations $\sqcap$ and $\sqcup$ defined on elements of S_E^+ as follows:

- $[\bot] \sqcap [C] = [\bot]$, and $[\top] \sqcap [C] = [C]$
- If $C_1, C_2 \in S$ have an *mgi* D then $[C_1] \sqcap [C_2] = [D]$ otherwise $[C_1] \sqcap [C_2] = [\bot]$
- $[\bot] \sqcup [C] = [C]$, and $[\top] \sqcup C] = [\top]$
- If C_1 and C_2 have *lgg* D then $[C_1] \sqcup [C_2] = [D]$ otherwise $[C_1] \sqcup [C_2] = [\top]$

The meet operation or glb called *mgi* stands for most-general-instance. If the set of positive literals in $C_1 \cup C_2$ have an mgu θ, then $mgi(C_1, C_2) = (C_1 \cup C_2)\theta$. Otherwise $mgi(C_1, C_2) = \bot$

The join operation or lub called *lgg* stands for least-general-generalisation of clauses (Lab Nos. 5, 6)

Example

$$S^{+} = \{ \Box, \bot,$$

$$is\_tiger(tom) \leftarrow has\_stripes(tom), is\_tawny(tom) \ ,$$

$$is\_tiger(bob) \leftarrow has\_stripes(bob), is\_white(bob) \ ,$$

$$is\_tiger(tom) \leftarrow has\_stripes(tom) \ ,$$

$$is\_tiger(tom) \leftarrow is\_tawny(tom) \ ,$$

$$is\_tiger(bob) \leftarrow has\_stripes(bob) \ ,$$

$$is\_tiger(bob) \leftarrow is\_white(tom) \ ,$$

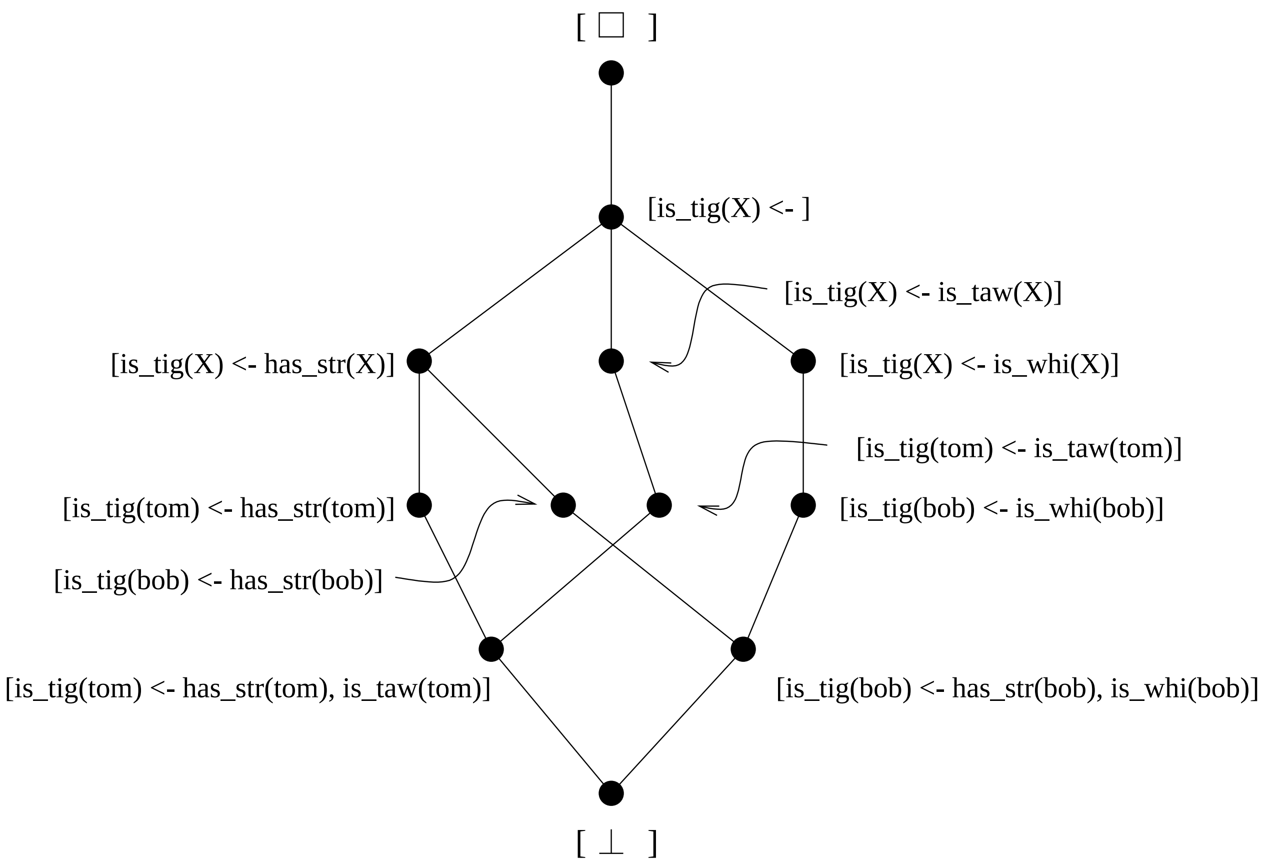
$$is\_tiger(X) \leftarrow has\_stripes(X) \ ,$$

$$is\_tiger(X) \leftarrow is\_tawny(X) \ ,$$

$$is\_tiger(X) \leftarrow is\_white(X) \ ,$$

$$is\_tiger(X) \leftarrow \}$$

Diagram of p.o. set S_E^+:



No Finite Chains in the Lattice

The existence of finite chains in lattices of atoms ordered by subsumption does *not* carry over to Horn clauses ordered by subsumption.

This follows from the observation that there are clauses which have no *finite* and complete set of downward covers

This makes it impossible to devise an ILP program that uses a refinement operator that is both complete and non-redundant

Relative Subsumption ordering over Horn clauses

Consider Horn clauses C, D and a set B:

$$D: \quad gfather(henry, john) \leftarrow$$

$$B: \quad \begin{array}{c} father(henry, jane) \leftarrow \\ father(henry, joe) \leftarrow \\ parent(jane, john) \leftarrow \\ parent(joe, robert) \leftarrow \end{array}$$

$$C: \quad gfather(X, Y) \leftarrow father(X, Z), parent(Z, Y)$$

Now $C \not\succeq D$. But $C \succeq D'$ where D':

$$gfather(henry, john) \leftarrow \begin{array}{l} father(henry, jane), \\ father(henry, joe), \\ parent(jane, john) \\ parent(joe, robert) \end{array}$$

Relative subsumption $C \succeq_B D$ if $C \succeq \bot(D, B)$ is a quasi-ordering

- $\bot(B, D)$ may not be Horn
- $\bot(B, D)$ may not be finite

Relative Subsumption Lattice over Horn clauses

Lattice only if B is a finite set of positive ground literals

Least upper bound of Horn clauses C_1, C_2

$$lgg_B(C_1, C_2) = lgg(\perp(B, C_1), \perp(B, C_2))$$

Greatest lower bound of Horn clauses C_1, C_2

$$glb_b(C_1, C_2) = glb(\perp(B, C_1), \perp(B, C_2))$$

The non-existence of finite chains in lattices of Horn clauses ordered by subsumption carries over to the lattice of clauses ordered by relative subsumption

Subsumption ordering over Horn clause-sets

Consider the set S of all finite Horn clause-sets in some language, and $S^+ = S \cup \{\bot\}$. Let $\succeq_\theta$ denote subsumption relation over Horn clauses and the dyadic relation $\succeq$ be such that:

- $\top = \{\Box\} \succeq T$ for all $T \in S^+$
- $T \succeq \bot$ for all $T \in S$
- $T_1 \succeq T_2$ iff $\forall D \in T_2\ \exists C \in T_1$ s.t. $C \succeq_\theta D$

$\succeq$ is a quasi-ordering known as "subsumption". A partial ordering results from the partition of S into the sets $\{\Box\}, X_1, \ldots$ where $[T]$ denotes all clause-sets that are subsume-equivalent to T. Two theories T_1,T_2 are subsume equivalent iff $T_1 \succeq T_2$ and $T_2 \succeq T_1$

Example of subsumption ordering on clause-sets

$$\{mem(A,[A|B]) \leftarrow, mem(A,[B,A|C]) \leftarrow\}$$

$$\succeq$$

$$\{mem(1,[1,2]) \leftarrow, mem(2,[1,2]) \leftarrow\}$$

Subsumption lattice of Horn clause-sets

It can be shown that the p.o. set of equivalence classes of Horn clause-sets S_E^+ is a lattice with the binary operations $\sqcap$ (glb) and $\sqcup$ (lub) defined on elements of S_E^+ (up to subsume-equivalence)

Given a pair $T_1, T_2 \in S_E^+$

$$lub(T_1, T_2) = T_1 \cup T_2$$

Given a pair $T_1, T_2 \in S_E^+$

$$glb(T_1, T_2) = \left\{ gs_{\mathcal{H}}(C_1', C_2') \;\middle|\; \begin{array}{l} \langle C_1, C_2 \rangle \in T_1 \times T_2 \\ \text{and } C_1', C_2' \text{ are variants} \\ \text{of } C_1, C_2 \text{ std. apart} \end{array} \right\}$$

where, using the definition mgi of Horn clauses

$$gs_{\mathcal{H}}(C_1, C_2) = \begin{cases} C_1 \cup C_2 & \text{if } C_1, C_2 \text{ headless} \\ mgi(C_1, C_2) & \text{otherwise} \end{cases}$$

No Finite Chains in the Lattice

The non-existence of finite chains in lattices of Horn clauses ordered by subsumption carries over to Horn clause-sets ordered by subsumption.

The implication ordering

In a manner analogous to subsumption, we can define a quasi-ordering based on implication between clauses (clause-sets)

$$C \succeq D \text{ if } C \models D$$

and a quasi-ordering based on relative implication

$$C \succeq_B D \text{ if } B \cup \{C\} \models D$$

The partial ordering over the resulting equivalence classes is not a lattice (lubs and glbs do not always exist)

Subsumption and Implication

The principal generality orderings of interest are subsumption ($\succeq_\theta$) and implication ($\succeq_{\models}$)

For clauses C, D, subsumption is *not* equivalent to implication

$$\text{if } C \succeq_\theta D \text{ then } C \succeq_{\models} D$$

but

not vice – versa

For example

$C: \; natural(s(X)) \leftarrow natural(X)$

$D: \; natural(s(s(X))) \leftarrow natural(X)$

The Subsumption Theorem

A key theorem linking subsumption and implication

> If Σ is a set of clauses and D is a clause, then $\Sigma \models D$ iff D is a tautology, or there exists a clause $D' \succeq_\theta D$ which can be derived from Σ using some form of resolution.

When Σ contains a single clause C then the only clauses that can be derived are the result of *self-resolutions* of C

Thus the difference between $C \succeq_{\models} D$ and $C \succeq_\theta D$ arises when C is self-recursive or D is tautological

Comparing Generality Orderings

Given a set of clauses S, clauses $C, D \in S$ and quasi-orders $\succeq_1$ and $\succeq_2$ on S, then $\succeq_1$ is *stronger* than $\succeq_2$ if $C \succeq_2 D$ implies $C \succeq_1 D$. If also for some $C, D \in S$ $C \not\succeq_2 D$ and $C \succeq_1 D$ then $\succeq_1$ is *strictly stronger* than $\succeq_2$

The implication ordering is strictly stronger than the subsumption ordering

Other Generality Orderings

Quasi-orders that are increasingly weaker can be devised from stronger ones. For example:

- $C \succeq_{\models} D$ iff $C \models D$

- $C \succeq_{\theta}$ iff there is a substitution θ s.t. $C \subseteq D$

- $C \succeq_{\theta'} D$ iff every literal in D is *compatible* to a literal in C and $C \succeq_{\theta} D$.

- $C \succeq_{\theta''} D$ iff $|C| \geq |D|$ and $C \succeq_{\theta'} D$

We would like the strongest ordering that is practical

Tractability

Logical implication between clauses is undecidable (even for Horn clauses)

Subsumption is decidable but NP-complete (even for Horn clauses)

Restrictions to the form of clauses can make subsumption efficient

– Determinate Horn clauses. There exists an ordering of literals in C and exactly one substitution θ s.t. $C\theta \subseteq D$

– $k - local$ Horn clauses. Partition a Horn clause into k "disjoint" sub-parts and perform k independent subsumption tests

More problems with $\models$

We have already looked at the lattice of clauses (quasi-)ordered by subsumption $\succeq_\theta$

The lattice structure implies the existence of $lubs$ (least generalisations) and $glbs$ (greatest specialisations) for pairs of clauses

The same is not true for the implication quasi-ordering $\succeq_{\models}$

Order	lub	glb
$\succeq_\theta$	$\surd$	$\surd$
$\succeq_{\models}$	$\times$	$\surd$

(for restricted languages $lubs$ for $\succeq_{\models}$ may well exist)

Practical Generality Ordering

The strongest quasi-order that is practical appears to be subsumption

Even that will require restrictions on the clauses being compared

Refinement Operators

Refinement operators are defined for a S with a quasi-ordering $\succeq$

- ρ is a *downward refinement operator* if $\forall C \in S : \rho(C) \subseteq \{D | D \in S \text{ and } C \succeq D\}$

- δ is an *upward refinement operator* if $\forall C \in S : \delta(C) \subseteq \{D | D \in S \text{ and } D \succeq C\}$

Desirable properties of ρ (and dually δ)

1. **Locally Finite.** $\forall C \in S$: $\rho(C)$ is finite and computable.

2. **Complete.** $\forall C \succ D$: $\exists E \in \rho^*(C)$ s.t. $E \sim D$

3. **Proper.** $\forall C \in S : \rho(C) \subseteq \{D | D \in S \text{ and } C \succ D\}$

Refinement Operators under $\succeq_\theta$

Example. With an equality theory $=/2$, $D \in \rho(C)$ if:

$$D = \begin{cases} p(X_1, X_2, \ldots, X_{n_p}) & \text{if } C = \Box \text{ and } p/n_p \in \mathcal{L} \\ & \text{and the } X_i \text{ are distinct} \\ C \cup \{\neg l\} & \text{otherwise} \end{cases}$$

where

$$l = \begin{cases} V = W & \text{where } V, W \text{ occur in } C \\ V = f(X_1, X_2, \ldots, X_{n_f}) & \text{where } V \text{ occurs in } C \\ & \text{and } f/n_f \in \mathcal{L} \text{ and} \\ & \text{the } X_i \text{ are distinct} \\ q(X_1, X_2, \ldots, X_{n_q}) & \text{where } q/n_q \in \mathcal{L} \\ & \text{and the } X_i \text{ occur in } C \end{cases}$$

There are no upward (downward) refinement operators that are locally finite, complete and proper for sets of clauses ordered by $\succeq_\theta$