Least generalization.

↳ Recall ... generalization of atoms

$$\left.\begin{matrix} P(f(a), g(c)) \\ P(f(b), x) \end{matrix}\right\} \rightarrow$$

$\{(a, \langle 1,2\rangle), (f(a), \langle 1\rangle), (c, \langle 2,2\rangle), (g(c), \langle 2\rangle)\}$

$\{(b, \langle 1,2\rangle), (f(b), \langle 1\rangle), (x, \langle 2\rangle)\}$

$(a, \ldots), (b, \ldots) \rightarrow z_1 \qquad (g(c), \ldots), (x, \ldots) \rightarrow z_2$

$$P(f(z_1), z_2)$$

↳ Now for clauses:..

Selection ① Find all pairs of compatible literals

Finite $\left\{ \begin{matrix} C = \{P(x), P(y), \ldots \neg P(a)\} \\ D = \{Q(b), P(a), \ldots \neg P(b)\} \end{matrix} \right.$

3 selections $= \{(\overset{h_1}{P(x)}, \overset{m_1}{P(a)}), \{\overset{l_2}{P(y)}, \overset{m_2}{P(a)}\}, \{\neg P(a), \neg P(b)\}\}$

Say selections:- $l_1 \quad l_2 \quad \quad l_n$

$m_1 \quad m_2 \; \cdots \; m_n$

Could be repetitions of l_i & m_i But not of (l_i, m_i) pairs

(2) Construct 2 compatible clauses

$$C_S = \bigvee(l_1, l_2 \ldots l_n)$$

$$D_S = \bigvee(m_1, m_2 \ldots m_n)$$

Treat these as atoms & anti-unify them

- $G = \{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\}$, for $\mathbf{l}_1 = P(f(a), f(x))$, $\mathbf{l}_2 = P(f(x), g(a))$, $\mathbf{l}_3 = Q(a)$ x
- $d = \{\mathbf{m}_1, \mathbf{m}_2\}$, for $\mathbf{m}_1 = P(f(b), x)$, $\mathbf{m}_2 = P(y, g(b))$. (Remember to stand apart)

then, the set of all selections of C and D can be ordered in the following sequence:

$$S = (\mathbf{l}_1, \mathbf{m}_1), (\mathbf{l}_1, \mathbf{m}_2), (\mathbf{l}_2, \mathbf{m}_1), (\mathbf{l}_2, \mathbf{m}_2)$$

leading to the constructing of the following clauses:

$$C_S = \vee(\mathbf{l}_1, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_2)$$

$$D_S = \vee(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_1, \mathbf{m}_2)$$

} Do not ignore orders

The clauses C_S and D_S are compatible, and have the following lub:

$$E_S = P(f(z_1), z_2) \vee P(z_3, z_4) \vee P(f(z_5), z_6) \vee P(z_7, g(z_1))$$

Theorem 28 *Let $\mathcal{C}$ be a clausal language. Let $C, D \in \mathcal{C}$ be clauses, and S be a sequence of all selections of C and D. Then an $lub(C_S, D_S)$ is an LGS of $\{C, D\}$.*

lub

More specifically: in above eg,

$E_S = lGS(C, D)$

Proof:-

A) $\underbrace{lub(C_s, D_s)}_{E_s} \succeq C$ & $\underbrace{lub(C_s, D_s)}_{E_s} \succeq D$

1. $E_s \succeq C_s$ $E_s \succeq D_s$ (By construction of atomic generalization)

2. Also $C_s \succeq C$ & $D_s \succeq D$

Why?

Because set of literals in C_s $\subseteq$ set of lits in C

3. $\therefore$ [A] holds

B) Let $F = \{\bar{l}_1, \ldots \bar{l}_m\}$ be s.t. $F \geq_1 C$ $F \geq_1 D$

To prove: $F \geq_1 E$ (E is just E_S reconstructed)

$$\begin{tikzcd}
 & F = \{\bar{l}_1 \ldots \bar{l}_m\} \arrow[dl, "\theta_2"'] \arrow[dr, "\theta_1"] & \\
D & & C
\end{tikzcd}$$

$\Rightarrow \forall\, l_i \in F,\ \bar{l}_i\theta_1 = l_i \in C$
$\bar{l}_i\theta_2 = m_i \in D$

$\Rightarrow S' = \{(l_1, m_1), (l_2, m_2), \ldots (l_m, m_m)\}$
is a selection for C & D

$\Downarrow$

$C_{S'} = \underline{\vee}(l_1, \ldots l_m)$ $\equiv \vee(\bar{l}_1 \ldots \bar{l}_m)\theta_1$

$D'_S = \underline{\vee}(m_1 \ldots m_m) \equiv \vee(\bar{l}_1 \ldots \ldots \bar{l}_m)\theta_2$

atomic lub

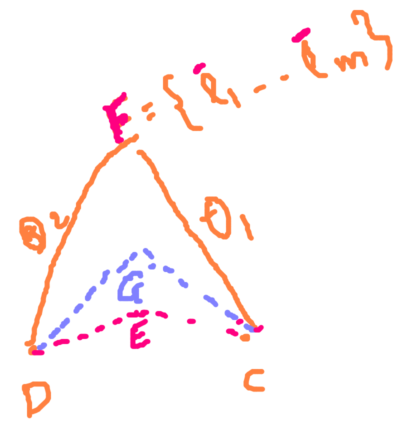
$G_S = \vee(k_1 \ldots k_m)$

$k_1\sigma_1 = l_1 = \bar{l}_1\theta_1$

$\exists\, \gamma$ s.t. $\vee(\bar{l}_1, \ldots \bar{l}_m)\gamma = \vee(k_1 \ldots k_m)$

$$\Rightarrow \vee(\underbrace{\bar{l}_1, \bar{l}_2 \ldots \bar{l}_m}_{F})\, r = \vee(\underbrace{k_1, r \ldots k_m}_{G})$$

$$\Rightarrow \quad F \succcurlyeq G$$



Note: Every selection in S' occurs in S_{all} $(S' \subseteq S_{all})$

we can prove that $\underset{S' \ldots}{G} \succcurlyeq \underset{r \ldots S}{E}$

$\underbrace{D, C}_{S_{all}} \rightarrow E \xrightarrow{S_{all}} E$

$F \rightarrow S'$

INPUT: Two clauses C and D;
OUTPUT: An LGS of $\{C, D\}$;
Let $(\mathbf{l}_1, \mathbf{m}_1), \ldots, (\mathbf{l}_n, \mathbf{m}_n)$ be a sequence of all selections of C and D;
Obtain $lub(\vee(\mathbf{l}_1, \ldots, \mathbf{l}_n), \vee(\mathbf{m}_1, \ldots, \mathbf{m}_n)) = \vee(\mathbf{l}_1, \ldots, \mathbf{l}_n)$ from the Anti-Unification Algorithm;
return $\{\mathbf{l}_1, \ldots, \mathbf{l}_n\}$;

Proof in figure.

$$\begin{tikzcd} N_1 \vee \ldots \vee N_m \arrow[d, "\gamma"'] \arrow[dr, "\theta_1 \text{ (resp. } \theta_2)"] & \\ G = K_1 \vee \ldots \vee K_m \arrow[r, "\sigma_1 \text{ (resp. } \sigma_2)"'] & \begin{array}{l} C_{S'} = L_1 \vee \ldots \vee L_m \\ (\text{resp. } D_{S'} = M_1 \vee \ldots \vee M_m) \end{array} \end{tikzcd}$$

If no selections $\rightarrow \square$ (empty clause) $= \top$

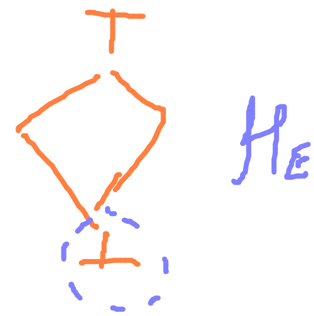
$C = Q(x) \vee P(a)$ $D = R(b) \vee S(f(a))$

$S_{all} \leq |C|\,|D|$

Q: What about Horn clauses?

Note:- You will need artificial "$\bot$"

Q: Do you need an artificial "$\top$"?



Thus, the p.o. set of equivalence classes of atoms $\mathcal{C}_E$ ($\mathcal{H}_E$) is a lattice with the binary operations $\sqcap$ and $\sqcup$ defined on elements of $\mathcal{C}_E$ ($\mathcal{H}_E$) as follows (note that $\perp \in \mathcal{C}$ is the empty set $\emptyset$):

- $[\perp] \sqcap [C] = [\perp]$, and $[\top] \sqcap [C] = [C]$
- If $C_1, C_2 \in \mathcal{C}$ ($C_1, C_2 \in \mathcal{H}$) have a GSS (glb) D then $[C_1] \sqcap [C_2] = [D] = [C_2\theta]$ otherwise $[C_1] \sqcap [C_2] = [\perp]$
- $[\perp] \sqcup [C] = [C]$, and $[\top] \sqcup C] = [\top]$
- If C_1 and C_2 have LGS (lub) M then $[C_1] \sqcup [C_2] = [M]$ otherwise $[C_1] \sqcup [C_2] = [\top]$

For $\mathcal{H}_E$ (horn clauses)

For $\mathcal{C}$,

- the GSS is simply the union $C_1 \cup C_1$ (union translates to 'or'ing the two clauses)
- the LGS of C_1 and C_2 is obtained using the LGS algorithm outlined in Figure 2.4

For C_E (normal clauses)

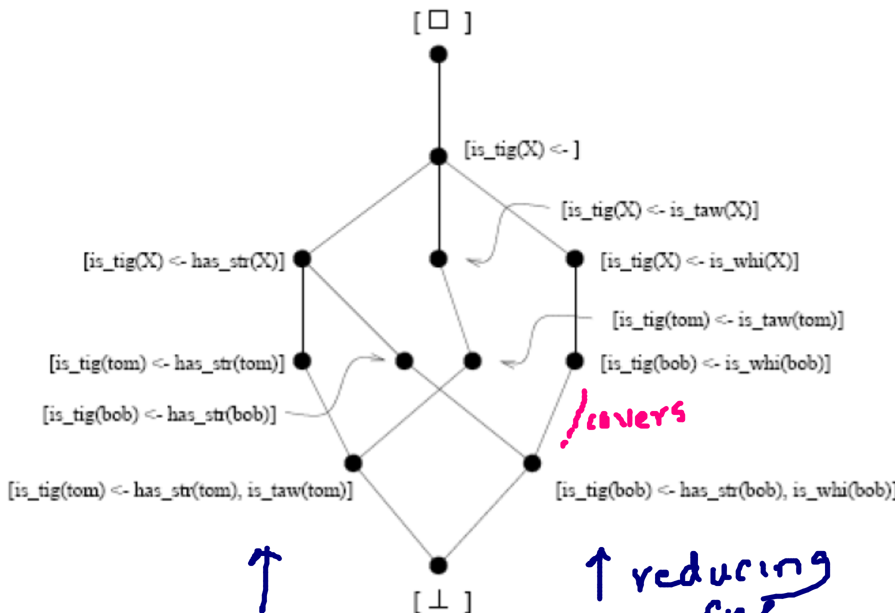
whereas, for $\mathcal{H}$,

- the GSS (or meet operation) is given as follows: If the set of positive literals (heads) in $C_1 \cup C_2$ have an mgu θ, then $GSS(C_1, C_2) = (C_1 \cup C_2)\theta$. Otherwise $GSS(C_1, C_2) = \perp$

$C_E \supseteq H_E$

- the LGS of C_1 and C_2 is obtained using the LGS algorithm outlined in Figure 2.4.

$$
\begin{aligned}
\mathcal{H} \;=\; \{\; & \Box,\ \bot, \\
& is\_tiger(tom) \leftarrow has\_stripes(tom), is\_tawny(tom)\ , \\
& is\_tiger(bob) \leftarrow has\_stripes(bob), is\_white(bob)\ , \\
& is\_tiger(tom) \leftarrow has\_stripes(tom)\ , \\
& is\_tiger(tom) \leftarrow is\_tawny(tom)\ , \\
& is\_tiger(bob) \leftarrow has\_stripes(bob) \\
& is\_tiger(bob) \leftarrow is\_white(tom)\ , \\
& is\_tiger(X) \leftarrow has\_stripes(X)\ , \\
& is\_tiger(X) \leftarrow is\_tawny(X)\ , \\
& is\_tiger(X) \leftarrow is\_white(X)\ , \\
& is\_tiger(X) \leftarrow \;\}
\end{aligned}
$$



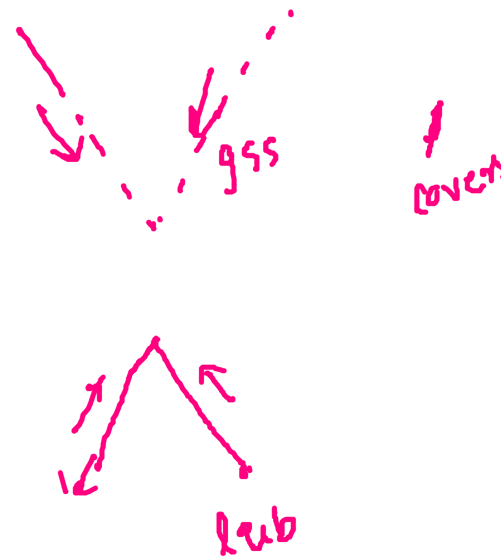
covers

↑ reduced no of constants

↑ reducing size

COVERS

- $\exists$ clauses which have no complete set of upward covers



- $\exists$ clauses with no upward covers.

$C = \{P(x_1, x_2)\}$

What abt $C_n = \{P(x_i, x_j) \mid i \neq j \text{ \& } 1 \leq i, j \leq n,\ n \geq 2\}$

$C_2 > C_3 > C_4 \cdots > C_n > C_{n+1} \cdots > C$

Similar negative result for downward cover.

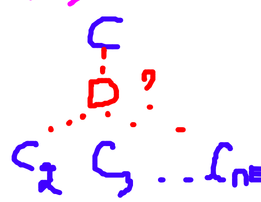
$$C: \{P(x_1, x_2), P(x_2, x_1)\}$$

$$E_n: \{P(y_1, y_2), P(y_2, y_3) \cdots P(y_{n-1}, y_n), P(y_n, y_1)\}, n \geq 2$$

$$C_n = C \cup E_n, \; n \geq 3$$

Then show that for any $n = 3^k$ $(k \geq 1)$

$$C \succ C_n$$



Refinement operator

ρ $\downarrow$ σ $\uparrow$

IDEAL

1. **Locally Finite:** $\forall C \in \mathcal{C}$: $\rho(C)$ is finite and computable. Otherwise ρ will be of limited use in practice.
2. **Complete:** $\forall C \succ D$: $\exists E \in \rho^*(C)$ s.t. $E \sim D$. That is, every specialization should be reachable by a finite number of applications of the operator.
3. **Proper:** $\forall C \in \mathcal{C} : \rho(C) \subseteq \{D | D \in S \text{ and } C \succ D\}$. That is, it is better only to compute proper generalizations of a clause, for otherwise repeated

Impossible to satisfy all three

You do not want to generate equivalent (reduced clauses)