5 Exercise session 5

Exercises on ILP.

5.1 Theta Subsumption

Order according to theta-subsumption.

1. $f(j,p) \leftarrow m(j), m(p), p(j,p).$
2. $f(X,Y) \leftarrow m(X), p(X,Y).$
3. $f(X,X) \leftarrow m(X), p(X,Y), p(X,Z).$
4. $f(X,X) \leftarrow m(X), p(X,X), m(X).$
5. $f(X,X) \leftarrow m(X), p(X,X).$
6. $f(X,X) \leftarrow m(X), p(X,Y).$

Solution

$$\begin{tikzcd}
 & 2 \arrow[dl, "\geq_\theta"'] \arrow[dr, "\geq_\theta"] & & 3 \sim 6 \arrow[dl, "\geq_\theta"'] \\
1 & & 4 \sim 5 &
\end{tikzcd}$$

5.2 LGG

Compute the LGG of the following literals and clauses:

a. $m(a, c(a, nil))$
   $m(b, c(c, nil))$

b. $add(s(s(0)), s(s(0)), s(s(s(s(0)))))$
   $add(s(0), s(0), s(s(0)))$

c. $m(X, c(X,Y)) \leftarrow c(X), list(Y).$
   $m(X, c(Y,Z)) \leftarrow c(X), c(Y), list(Z), m(X,Z).$

Solution

a. $m(X, c(Y, nil))$

b. $add(s(X), s(X), s(s(Y)))$

c. $m(R, c(S,T)) \leftarrow c(R), c(S), list(T).$

Step-by-step solution: 5.2a

Refer to the Slide 3 for this exercise session for the LGG rules.

$$\textbf{LGG}\left(m(a, c(a, nil));\ m(b, c(c, nil))\right) =$$

$$\boxed{\text{First rule in } \textit{LGG of literals}\text{: two instances of the same predicate}}$$

$$m(\textbf{LGG}\left(a;\ b\right), \textbf{LGG}\left(c(a, nil);\ c(c, nil)\right)) =$$

$$\boxed{\text{Second rule in } \textit{LGG of terms}\text{: two different terms are generalized into variable}}$$

$$\boxed{\text{It is important to keep track of such variables, e.g. by using subscripts}}$$

$$m(V_{ab}, \textbf{LGG}\left(c(a, nil);\ c(c, nil)\right)) =$$

$$\boxed{\text{First rule in } \textit{LGG of terms}\text{: two instances of the same functor}}$$

$$m(V_{ab}, c(\textbf{LGG}\left(a;\ c\right), \textbf{LGG}\left(nil;\ nil\right))) =$$

$$\boxed{\text{Second rule in } \textit{LGG of terms}}$$

$$m(V_{ab}, c(V_{ac}, \textbf{LGG}\left(nil;\ nil\right))) =$$

$$\boxed{\text{First rule in } \textit{LGG of terms}\text{: a constant is a functor without arguments}}$$

$$m(V_{ab}, c(V_{ac}, nil)) =$$

$$\boxed{\text{Renaming variables (not necessary)}}$$

$$m(X, c(Y, nil))$$

Answer: $\textbf{LGG}\left(m(a, c(a, nil));\ m(b, c(c, nil))\right) = m(X, c(Y, nil))$
Interpretation: We knew (e.g. from the data) that the property m held (separately) for objects a and b paired with other *specific* objects described by the functor c with certain arguments. By applying the LGG operator, we conjecture that m holds for all objects (variable X can be substituted by any domain object) paired with a larger number of other objects (the functor c now has a variable Y as its argument). This conjecture might be wrong, e.g. not hold in the available data.

Step-by-step solution: 5.2c

$$\mathbf{LGG}\left(m(X,c(X,Y)) \leftarrow c(X), \mathit{list}(Y);\ m(X,c(Y,Z)) \leftarrow c(X), c(Y), \mathit{list}(Z), m(X,Z)\right) =$$

$$\boxed{\text{Convert implications into sets of literals}}$$

$$\mathbf{LGG}\left(\{m(X,c(X,Y)), \neg c(X), \neg \mathit{list}(Y)\};\ \{m(X,c(Y,Z)), \neg c(X), \neg c(Y), \neg \mathit{list}(Z), \neg m(X,Z)\}\right) =$$

$$\boxed{\text{Rule } \textit{LGG of clauses}\text{: LGGs of all literal pairs that are defined}}$$

$$\boxed{\text{Note that it is a one long set, up until the next equals sign}}$$

$$\boxed{\text{All pairs for the first literal in the first clause}}$$

$$\{\mathbf{LGG}\left(m(X,c(X,Y);\ m(X,c(Y,Z))\right); \mathbf{LGG}\left(m(X,c(X,Y);\ \neg c(X)\right); \mathbf{LGG}\left(m(X,c(X,Y);\ \neg c(Y)\right);$$
$$\mathbf{LGG}\left(m(X,c(X,Y);\ \neg \mathit{list}(Z)\right); \mathbf{LGG}\left(m(X,c(X,Y);\ \neg m(X,Z)\right);$$

$$\boxed{\text{All pairs for the second literal in the first clause}}$$

$$\mathbf{LGG}\left(\neg c(X);\ m(X,c(Y,Z))\right); \mathbf{LGG}\left(\neg c(X);\ \neg c(X)\right); \mathbf{LGG}\left(\neg c(X);\ \neg c(Y)\right);$$
$$\mathbf{LGG}\left(\neg c(X);\ \neg \mathit{list}(Z)\right); \mathbf{LGG}\left(\neg c(X);\ \neg m(X,Z)\right);$$

$$\boxed{\text{All pairs for the third literal in the first clause}}$$

$$\mathbf{LGG}\left(\neg \mathit{list}(Y);\ m(X,c(Y,Z))\right); \mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg c(X)\right); \mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg c(Y)\right);$$
$$\mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg \mathit{list}(Z)\right); \mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg m(X,Z)\right)\} =$$

$$\boxed{\text{Removing all LGGs that are undefined}}$$

$$\{\mathbf{LGG}\left(m(X,c(X,Y);\ m(X,c(Y,Z))\right); \cancel{\mathbf{LGG}\left(m(X,c(X,Y);\ \neg c(X)\right)}; \cancel{\mathbf{LGG}\left(m(X,c(X,Y);\ \neg c(Y)\right)};$$
$$\cancel{\mathbf{LGG}\left(m(X,c(X,Y);\ \neg \mathit{list}(Z)\right)}; \cancel{\mathbf{LGG}\left(m(X,c(X,Y);\ \neg m(X,Z)\right)};$$
$$\cancel{\mathbf{LGG}\left(\neg c(X);\ m(X,c(Y,Z))\right)}; \mathbf{LGG}\left(\neg c(X);\ \neg c(X)\right); \mathbf{LGG}\left(\neg c(X);\ \neg c(Y)\right);$$
$$\cancel{\mathbf{LGG}\left(\neg c(X);\ \neg \mathit{list}(Z)\right)}; \cancel{\mathbf{LGG}\left(\neg c(X);\ \neg m(X,Z)\right)};$$
$$\cancel{\mathbf{LGG}\left(\neg \mathit{list}(Y);\ m(X,c(Y,Z))\right)}; \cancel{\mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg c(X)\right)}; \cancel{\mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg c(Y)\right)};$$
$$\mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg \mathit{list}(Z)\right); \cancel{\mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg m(X,Z)\right)}\} =$$

$$\{\mathbf{LGG}\left(m(X,c(X,Y);\ m(X,c(Y,Z))\right); \mathbf{LGG}\left(\neg c(X);\ \neg c(X)\right); \mathbf{LGG}\left(\neg c(X);\ \neg c(Y)\right); \mathbf{LGG}\left(\neg \mathit{list}(Y);\ \neg \mathit{list}(Z)\right)\} =$$

$$\boxed{\text{Proceeding with applying the LGG operator}}$$

$$\{m(X, c(\mathbf{LGG}\left(X;\ Y\right), \mathbf{LGG}\left(Y;\ Z\right)); \neg c(X); \neg c(\mathbf{LGG}\left(X;\ Y\right)); \neg \mathit{list}(\mathbf{LGG}\left(Y;\ Z\right))\} =$$
$$\{m(X, c(V_{XY}, V_{YZ}); \neg c(X); \neg c(V_{XY}); \neg \mathit{list}(V_{YZ})\} =$$

$$\boxed{\text{Converting a set of clauses back into an implication}}$$

$$m(X, c(V_{XY}, V_{YZ}) \leftarrow c(X), c(V_{XY}), \mathit{list}(V_{YZ}) =$$

$$\boxed{\text{Renaming introduced variables (not necessary)}}$$

$$m(R, c(S,T)) \leftarrow c(R), c(S), \mathit{list}(T)$$

Answer:
$\mathbf{LGG}\left(m(X,c(X,Y)) \leftarrow c(X), \mathit{list}(Y);\ m(X,c(Y,Z)) \leftarrow c(X), c(Y), \mathit{list}(Z), m(X,Z)\right) = m(R, c(S,T)) \leftarrow c(R), c(S), \mathit{list}(T)$

5.3 RLGG

Compute the RLGG of:

$gf(a,c)$
$gf(b,d)$

according to the following background:

$p(a,b)$, $p(b,c)$, $p(c,d)$, $m(a)$, $m(b)$, $f(c)$, $m(d)$

Solution

$rlgg(e_1, e_2) = lgg(e_1 \leftarrow B, e_2 \leftarrow B) = lgg(c_1, c_2)$ with:

$c_1 : gf(a, c) \leftarrow p(a, b), p(b, c), p(c, d), m(a), m(b), f(c), m(d)$
$c_2 : gf(b, d) \leftarrow p(a, b), p(b, c), p(c, d), m(a), m(b), f(c), m(d)$

Applying $lgg(c_1, c_2) = \{lgg(l_1, l_2) \mid l_1 \in c_1 \wedge l_2 \in c_2 \wedge lgg(l_1, l_2)$ defined$\}$ we get:

$$
\begin{array}{llll}
gf(A,B) \leftarrow & p(a,b), p(A,C), p(D,E), & m(a), m(A), m(K), & f(c) \\
& p(F,G), p(b,c), p(C,B), & m(F), m(b), m(E), & \\
& p(H,I), p(G,J), p(c,d), & m(L), m(I), m(d) &
\end{array}
$$

with:

$\theta_1 = \{A/a, B/c, C/b, D/a, E/b, F/b, G/c, H/c, I/d, J/d, K/a, L/d\}$
$\theta_2 = \{A/b, B/d, C/c, D/c, E/d, F/a, G/b, H/a, I/b, J/c, K/d, L/a\}$

Reordering of the literals gives:

$$
\begin{array}{ll}
gf(A,B) \leftarrow & p(A,C), m(A), p(C,B), \\
& p(D,E), m(E), \\
& p(F,G), m(F), p(G,J), \\
& p(H,I), m(I), \\
& m(K), m(L), \\
& p(a,b), p(b,c), p(c,d), m(a), m(b), f(c), m(d).
\end{array}
$$

The literals on the middle four lines can be removed because the clause remains equivalent under θ-subsumption. Consider for example the last literal $m(L)$. Call c_a the clause with this literal and c_b the clause without this literal. We have to show that $c_a \sim c_b$ or that $c_a \leq_\theta c_b$ and $c_b \leq_\theta c_a$. For the first condition we apply $\theta = \{L/A\}$. The second condition is trivial. It is also possible to use atoms listed in the background in the substitution, e.g. $\theta = \{L/a\}$.

Facts from the background knowledge are true for all examples and have to be removed from the obtained clause after the reduction described above (as described in the definition of RLGG).

The result is $gf(A, B) \leftarrow p(A, C), m(A), p(C, B)$. This is exactly the definition of the grandfather predicate.

5.4 Using RLGG

Learn a definition for the mortal/1 predicate by using RLGG and the following learning examples:

$+ : mortal(socrates)$
$- : mortal(dracula)$
$+ : mortal(bobby)$

Background:

$man(socrates)$, $alive(socrates)$, $bat(dracula)$, $dead(dracula)$, $dog(bobby)$, $alive(bobby)$.

Solution

$mortal(X) \leftarrow alive(X).$

5.5 Inverse Resolution

Find an inverse resolution path that derives the clause

$daughter(X, Y) : -parent(Y, X), female(X).$

starting from

> $daughter(lisa, marge).$
> $parent(marge, lisa).$
> $female(lisa).$

Solution

Multiple solutions are possible, see one example below. This scheme should be read bottom-up, like the inverse resolution schemes on the slides.

Note that in both cases *absorption* is applied (a new condition is "absorbed" into the body of the constructed rule) and that variables can be introduced via *inverse substitution*.

daughter(X,Y) $\leftarrow$ parent(Y,X), female(X).
daughter(lisa,Y) $\leftarrow$ parent(Y,lisa), female(lisa).   female(lisa).

daughter(lisa,Y) $\leftarrow$ parent(Y,lisa).

daughter(X,Y) $\leftarrow$ parent(Y,X).
daughter(lisa,Y) $\leftarrow$ parent(Y,lisa).
daughter(X,marge) $\leftarrow$ parent(marge,X).
daughter(lisa,marge) $\leftarrow$ parent(marge,lisa).   parent(marge,lisa).

daughter(lisa,marge).