1. Consider two atoms $A_1 = P(t_1, \ldots, t_n)$ and $A_2 = P(s_1, \ldots, s_n)$, where the $t_i$ and $s_i$ are terms. Let $A = \text{LGG}(A_1, A_2)$, where LGG is the procedure described in class for computing least-general generalizations under $\theta$-subsumption. Prove that $A$ $\theta$-subsumes both $A_1$ and $A_2$. That is, show that $A$ is indeed a generalization of the inputs.

2. Recall that for two distinct constants $c$ and $c'$ we defined $\text{LGG}(c, c') = V_{c,c'}$, where $V_{c,c'}$ is a variable that is indexed by the pair $(c, c')$. Consider a new version of LGG which we will call $\text{LGG}'$. $\text{LGG}'$ is identical to LGG except that for pairs of distinct constants $c$ and $c'$ we have $\text{LGG}'(c, c') = V$, where $V$ is a new variable that has not been introduced anywhere else in the LGG' execution. Show that in general $\text{LGG}'$ does not compute a least-general generalization under $\theta$-subsumption. In particular show two clauses (possibly just two atoms) such that $\text{LGG}'$ produces a clause that is strictly more general under $\theta$-subsumption than the clause produced by LGG.

3. Consider the two clauses:

\[
c_1 = P(x) \rightarrow P(f(x))
\]
\[
c_2 = P(x) \rightarrow P(f(f(x)))
\]

Prove that $c_1 \models c_2$, but that $c_1$ does not $\theta$-subsume $c_2$. Note that to show that $c_1 \models c_2$ it is sufficient to show that $c_2$ can be derived/proved from a knowledge base that contains just $c_1$.

4. Consider the following learning problem with background knowledge $B$, positive examples $P$, and negative examples $N$.

\[
P = \{ Q(a, b), Q(d, e), Q(g, h) \}
\]
\[
N = \{ Q(j, k) \}
\]
\[
B = \{ R(a, b), R(b, c), R(d, e), R(e, f), R(g, h), P(h), R(j, k), P(j) \}
\]

Simulate running Golem on this learning problem to learn a rule set that covers all positives but no negatives. For pruning LGGs you can use the pruning technique that removes literals from the body if the removal does not result in covering negative examples.