1. Recall that a *definite clause* is an atom (or fact) or a rule of the form

\[ h : - \ b_1, b_2, \ldots, b_m \]

where \( h \) and all \( b_i \)'s are atoms. A *propositional clause* is a definite clause in which all predicates in it have arity 0. (That is, there are no terms.) Let us agree to encode propositional clauses as lists, with an atom \( f \) encoded as \([f]\) and a rule \( h : - b_1, \ldots, b_m \) as \([h, b_1, \ldots, b_m]\). A finite list of propositional clauses can then be encoded as a list of lists — e.g.

\[
\begin{align*}
  h &: - c. \\
  h &: - f, g. \\
  f &: - g. \\
  g &:
\end{align*}
\]

as \([[h, c], [h, f, g], [f, g], [g]]\). Recall that the binary Prolog predicate \( \text{lc} (\text{Atom}, \text{KB}) \) defined below is true iff \( \text{Atom} \) is a logical consequence of \( \text{KB} \).

\[
\text{lc}(X, \text{KB}) : - \ \text{cn}(C, \text{KB}), \ \text{member}(X, C).
\]

\[
\text{cn}(C, \text{KB}) : - \ \text{cn}([], C, \text{KB}).
\]

\[
\text{cn}(\text{TempC}, C, \text{KB}) : - \ \text{member}([H|B], \text{KB}), \\
  \text{all}(B, \text{TempC}), \\
  \text{nonmember}(H, \text{TempC}), \\
  \text{cn}([H|\text{TempC}], C, \text{KB}).
\]

\[
\text{cn}(C, C, \_).
\]

\[
\text{all}([], \_).
\]

\[
\text{all}([H|T], L) : - \ \text{member}(H, L), \ \text{all}(T, L).
\]

\[
\text{nonmember}(\_, [\_]).
\]

\[
\text{nonmember}(X, [H|T]) : - X=H, \ \text{nonmember}(X, T).
\]

Your task is to define a predicate \( \text{lcRule} (\text{List}, \text{KB}) \) that is true precisely when the rule encoded by \( \text{List} \) is a logical consequence of \( \text{KB} \).

**Some runs to cover**

\[
| ?- \ \text{lcRule}([\text{a}, \text{b}],[\text{a}],[\text{b}, \text{c}])).
\]

yes

\[
| ?- \ \text{lcRule}([\text{b}, \text{a}],[\text{a}],[\text{b}, \text{c}])).
\]
| no |
| ?- lcRule([a,b],[[a,b,c],[c]]). yes |
| no |
| ?- lcRule([a,d],[[a,b,c],[c]]). yes |

2. Consider the knowledge base

\[
\begin{align*}
\text{false} & : \neg a. \\
\text{false} & : \neg b, c. \\
a & : \neg d. \\
b & : \neg e. \\
c & : \neg d, f. \\
c & : \neg g. \\
c & : \neg h. 
\end{align*}
\]

Given that \(\{d, e, f, g, h\}\) is the set of assumables, what is the set of minimal conflicts of the above clauses?