Exercise 1 (Avoiding Recursion Caused by Magic Sets). Consider the following rule set:

Rules R:

\[ b(X) \leftarrow d_1(X) \]
\[ b(X) \leftarrow d_2(X) \]
\[ p(X) \leftarrow c(X, Y), \neg b(X), b(Y) \]

a) Show that R is hierarchical and stratifiable.

b) How many admissible SIPS are there?

c) Show that the Magic Sets transformed rules for the query \( p(1) \) are either unstratifiable or non-hierarchical.

d) Try to avoid the instruction of recursion (which would solve the stratification problem at the same time) using the RACS and the CSE algorithm.

Exercise 2 (Properties of Update Rules). Consider the following rules:

\[ p(X, Y) \leftarrow e(X, Y), \text{not } b(X, Y). \]
\[ p(X, Y) \leftarrow e(X, Z), p(Z, Y). \]

Let us suppose the database contains the facts \( e(1,2) \) and \( e(2,1) \), only. The deletion of \( e(2,1) \) would cause the derived facts \( p(2,1), p(2,2) \) and \( p(1,2) \) to be deleted, as well. Show that the following propagation rules:

\[ p^-(X, Y) \leftarrow e^-(X, Y), \text{not } b^{old}(X, Y), \text{not } p^{new}(X, Y). \]
\[ p^-(X, Y) \leftarrow b^+(X, Y), e^{old}(X, Y), \text{not } p^{new}(X, Y). \]
\[ p^-(X, Y) \leftarrow e^+(X, Z), p^{old}(Z, Y), \text{not } p^{new}(X, Y). \]
\[ p^-(X, Y) \leftarrow p^+(Z, Y), e^{old}(X, Z), \text{not } p^{new}(X, Y). \]

\[ p^{new}(X, Y) \leftarrow p^{old}(X, Y), \text{not } p^-(X, Y). \]
\[ e^{new}(X, Y) \leftarrow e^{old}(X, Y), \text{not } e^-(X, Y). \]
\[ e^{new}(X, Y) \leftarrow e^+(X, Y). \]
\[ b^{new}(X, Y) \leftarrow b^{old}(X, Y), \text{not } b^-(X, Y). \]
\[ b^{new}(X, Y) \leftarrow b^+(X, Y). \]

do not correctly determine the induced changes. Why are the rules incorrect?
**Exercise 3** (Update Propagation). Consider the following database:

\[
\begin{align*}
p(X, Y) &\leftarrow e(X, Y), \textbf{not } b(X, Y). \\
p(X, Y) &\leftarrow e(X, Z), p(Z, Y).
\end{align*}
\]

\[
e(1,2). \ e(2,3). \ e(4,5). \ e(5,6). \ e(6,7). \ +b(2,3)!
\]

\[
e(7,8). \ e(7,9). \ e(8,10). \ e(9,10). \ e(10,11). \ e(11,12). \ e(12,13). \ +e(2,20)! +e(20,3)!
\]

Show the benefits of using incremental update propagation rather than naive change computation by identifying relevant facts for determining the induced changes to p.