Pseudo code for Possibilistic database Schema to remove data redundancy

Input :

- Relational Schema(R) , Possibility Degree (or) Certainty degree Scale(S), set of possibilistic functional dependencies.

Algorithm :

For β-BCNF :

- For each $beta_i$(pfds of one world) :
    - Compute canonical cover of pfds (R_1)
    - define D (empty set of relations), LS (empty set of pfds)
    - Check whether pfds in cover is in bcnf form or not
    - For each pfd in Σ:
        - If (violates bcnf) :
            - Append given pfd to LS.
    - do
        - For every pfd in LS:
            - Perform an αk+1-i-lossless BCNF decomposition for a pFD set Σ i.e, by performing a classical lossless BCNF decomposition for the βi-cut Σβi of Σ.
            - Append obtained decompositions to D,remove given pfd from LS and recursively check if D is in bcnf
    - While (LS is not empty)
    - Remove pfds which are there in LS from R_1.
    - Output (R_1 + D(Set of decompositions))

For β-3NF :

- For each $beta_i$(pfds of one world) :
  - Compute canonical cover of pfds (R_1)
  - define D (empty set of relations), LS (empty set of pfds)
  - Check whether pfds in cover is in 3-NF form or not
  - For each pfd in Σ:
    - If (violates 3-NF) :
      - Append given pfd to LS.
  - Do
    - For every pfd in LS:
      - Perform (αk+1-i)-lossless,βi-dependency-preserving 3NF synthesis for a pFD set Σ i.e, by performing a classical lossless 3NF synthesis for Σβi.
      - Append obtained decompositions to D,remove given pfd from LS and recursively check if D is in 3nf
  - While (LS is not empty)
  - Remove pfds which are there in LS from R_1.
  - Output (R_1 + [pfd for every pfd in LS]) (list of decomposed relations)