```
IDF ComputeIteratedDominanceFrontier(G_ac:Acyclic Graph, X: subset of V(G) )
{
   ItDomFtr = { }
   for every vertex u of V(G) – {Entry(G)} in topological sort order do {
      ReachingDefs = { }
      for every predecessor v of u do {
         if v ε ( ItDomFtr U X U {Entry(G)} ) {
            ReachingDefs = ReachingDefs U {v}
         else
            ReachingDefs = ReachingDefs U UniqueReachingDefs(v)
         } //endif
      } //end for

      if ( |ReachingDefs| = 1 ) {
         UniqueReachingDefs(u) = only element of ReachingDefs
      else
         ItDomFtr = ItDomFtr U {u}
      } //endif

   }  //end for
   return ItDomFtr
   }
}
```