/*REBAC POLICY MINING FEASIBILITY DETECTION VERSION 1*/

/*PROGRAMMING ENVIRONMENT: ECLIPSE JAVASE-9*/

/*Motive: Given EAS and Relationship Graph (RG), is the ReBAC rule generation feasible? */

****
PROGRAM INPUT
****
This program needs three input files as follows:

1) isolatedVertex.txt

If the RG contains any isolated vertex, it must be noted in the file, one in each line.
Example
****
If set of vertices in RG, V = {A, B, C, D}, set of edges E = {(A, B, F)}, and set of relation type specifier Σ={F}, then this file contains two lines:

C
D

If there exists no isolated vertex, the file will remain empty.

**Vertex is a string

2) Edges.txt
This file contains directed labeled edge information of RG, one edge in each line in the following format:

originating-vertex terminating-vertex edge-direction edge-label

**originating-vertex, terminating-vertex, edge-direction, and edge-label are strings
**For now, edge-direction is restricted to be directed only, represented by "D".

Example
****
If set of vertices in RG, V = {A, B, C, D}, set of edges E = {(A, B, F), (B, C, F)}, and set of relation type specifier Σ={F}, then this file contains two lines:

A B D F
B C D F

3) EAS.txt
This file contains authorization relation (AUTH), one tuple in each line in the following format:

originating-vertex terminating-vertex

where originating-vertex is authorized to perform operation op on terminating-vertex.

**originating-vertex and terminating-vertex are strings

Example
****
If set of vertices in EAS, V = {A, B, C, D}, then the file containing two lines:

A B
B C

indicates that A is authorized to perform operation op on B, and B is authorized to perform operation op on C.

****
PROGRAM OUTPUT
****
This program needs generates two output files as follows:

1) finalRG.txt

This file contains the list of vertices and list of edges in RG.

Example
****
If set of vertices in RG, V = {A, B, C, D}, set of edges E = {(A, B, F)}, and set of relation type specifier Σ={F}, then this file output is as follows:
List of vertices
A B C D
List of edges
A B F

2) finalRuleOutput.txt
This file contains the generated rule, feasible/infeasible status, and rule minimization messages. In case of infeasibility, all infeasible authorization tuples are noted as well.

****
CASE EXAMPLES WITH OUTPUT
****
Total six cases have been provided as example. Amongst them, four cases

have been analyzed using Simple Path(SP), Simple Complementary Path(SCP), Simple Permissive Path (SPP), and Simple Complementary Permissive Path (SCPP). Two other randomly generated cases are analyzed with SP only.

****
SPECIAL NOTE
****

i) enhanceRG(enhanceType,vertexList,edgeList);
- -Enhances graph with complementary path when enhanceType=1
- -Enhances graph with permissive path when enhanceType=2
- -Enhances graph with complementary permissive path when enhanceType=3

By default, enhanceType=0, which indicates no change in the given RG.

ii) Currently, relationship type specifiers in RG are bounded to the given edge labels only.

iii) Program will be improved to manage large volume of input.

iv) If RG is enhanced, rule may contain non-relationship, inverse, and non-relationship inverse relationship labels. Given a relation, say F, non-relationship, inverse, and non-relationship inverse relationship labels are represented by F!, F^, and F!^, respectively.

/***/

This is version 1 of this program.

/***/