ARTIFICIAL INTELLIGENCE

PROJECT 3 - REPORT

(Role Based Access Control)

Lakshmi Narayanan Rajendran (112046482)
Abhinav Srivastava (112129749)

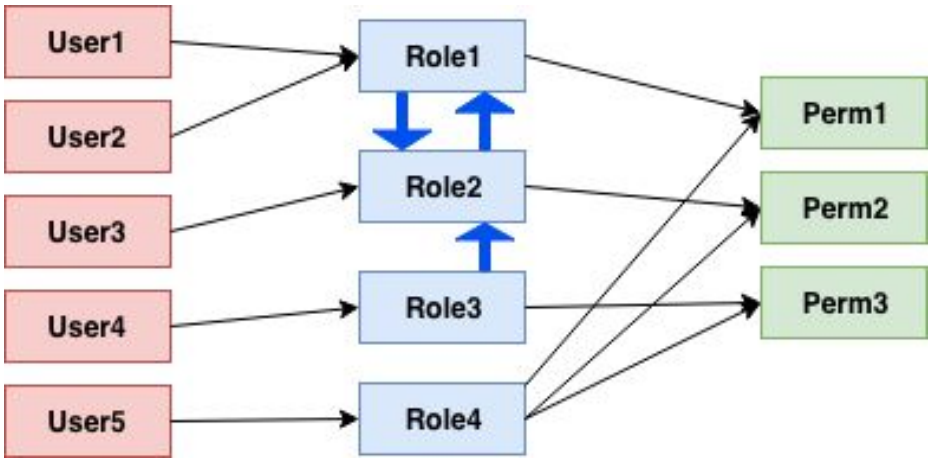
FILES

project3.pl - Contains all the method definitions to calculate the predicates
test.pl - Contains input (facts) and output format

Note: While loading the files into xsb, load **test.pl** first and **project3.pl** next.

FACTS:

users(5).
roles(4).
perms(3).
ur(1,1).
ur(2,1).
ur(3,2).
ur(4,3).
ur(5,4).
rh(1,2).
rh(2,1).
rh(2,3).
rp(1,1).
rp(2,2).
rp(3,3).
rp(4,1).
rp(4,2).
rp(4,3).



---

authorized_roles(User,List_Roles)

RESULTS:

?- authorized_roles(1,R).
R= [1,2,3]
Analysis:
User1 has one direct role **Role1**.
Role1 has one descendant **Role 2**.
Role2 has one descendant **Role3**.
Role2 also has descendant **Role1**, but is ignored because **Role1** is already visited.
Hence, **User1** has [**Role1, Role2, Role3**] as authorized roles.

?- authorized_roles(2,R).
R = [1,2,3]
Analysis:
User2 has one direct role **Role1**.
Role1 has one descendant **Role2**.
Role2 has one descendant **Role3**.
Role2 also has descendant **Role1**, but is ignored because **Role1** is already visited.
Hence, **User2** has [**Role1, Role2, Role3**] as authorized roles.

?- authorized_roles(3,R).
R = [1,2,3]

Analysis:
User3 has one direct role **Role2.**
Role2 has one descendant **Role3.**
Role2 has one descendant **Role1.**
Role1 has one descendant **Role2**, but is ignored because **Role2** is already visited.
Hence, **User3** has [**Role1, Role2, Role3**] as authorized roles.

?- authorized_roles(4,R).
R = [3]
Analysis:
User4 has one direct role **Role3**.
Role3 has no descendants.
Hence, **User4** has [**Role3**] as authorized roles.

?- authorized_roles(5,R).
R = [4]
Analysis:
User5 has one direct role **Role4**.
Role4 has no descendants.
Hence, **User5** has [**Role4**] as authorized roles.

---

authorized_permissions(User,List_Permissions)

RESULTS:

?- authorized_permissions(1,P).
P = [1,2,3]
Analysis:
User1 has one direct role **Role1** and two descendents roles as **Role2** and **Role3**.
Role1 has one direct permission **Permission1**.
Role2 has one direct permission **Permission2**.
Role3 has one direct permission **Permission3**.
So, **Role1** will inherit the permissions from **Role2** and **Role3**.
Hence, **User1** has [**Permission1, Permission2, Permission3**] as authorized permissions.

?- authorized_permissions(2,P).
P = [1,2,3]

Analysis:
User2 has one direct role **Role1** and two descendents roles as **Role2** and **Role3**.
Role1 has one direct permission **Permission1**.
Role2 has one direct permission **Permission2**.
Role3 has one direct permission **Permission3**.
So, **Role1** will inherit the permissions from **Role2** and **Role3**.
Hence, **User2** has [**Permission1, Permission2, Permission3**] as authorized permissions.

?- authorized_permissions(3,P).
P = [1,2,3]
Analysis:
User3 has one direct role **Role2** and two descendents roles as **Role1(cyclic)** and **Role3**.
Role2 has one direct permission **Permission2**.
Role1 has one direct permission **Permission1**.
Role3 has one direct permission **Permission3**.
So, **Role2** will inherit the permissions from **Role1** and **Role3**.
Hence, **User3** has [**Permission1, Permission2, Permission3**] as authorized permissions.

?- authorized_permissions(4,P).
P = [3]
Analysis:
User4 has one direct role **Role3** and no descendents roles.
Role3 has one direct permission **Permission3**.
Hence, **User4** has [**Permission3**] as authorized permissions.

?- authorized_permissions(5,P).
P = [1,2,3]
Analysis:
User5 has one direct role **Role4** and no descendents roles.
Role4 has three direct permissions **Permission1, Permission2, Permission3.**
Hence, **User5** has [**Permission1, Permission2, Permission3**] as authorized permissions.

---

minRoles(S)

RESULTS:

?- minRoles(S).
S = 2.

Analysis:

User1, User2, User3 and User5 have same permissions **[Permission1,Permission2,Permission3].**
So, these four users can be defined by a single role having three permissions [**Permission1, Permission2, Permission3**]

User4 has only one permission **[Permission3].**
So we define a new role with one permission as **[Permission3].**

Hence, there are **two** minimum roles required to cover all the users.