**Motivation and Background**

**Motivation**
- Deploy of RBAC requires identification of a complete set of roles.
- Role engineering - one of the costliest tasks in migrating to RBAC.
- Most of the existing approaches do not consider roles that already exist.

**Contributions**
- Formally define the problem of mining role hierarchy with minimal perturbation
- Present a heuristic algorithm to find an RBAC state as similar as possible to the existing state and the optimal state

**Background**
Role Based Access control (RBAC) - a high level authorization model in which access decisions are based on the roles that users hold within an organization. RBAC is a typical choice for organizational access control.

**The Problem of Mining Role Hierarchy with Minimal Perturbation**

The reduced concept lattice defines a complete RBAC state.
- Each concept represents a role
- The lattice can be viewed as the role hierarchy
- The sub-concept relation corresponds to the role inheritance relation.

We need a measure to compare the different role hierarchies and identify which one is more desirable.

We define two different measure:

A measure for Goodness of an RBAC State

Given \( W = \{w_1, w_2, w_3, w_4\} \) where \( w_1, w_2, w_3, w_4 \in G \cup \{\emptyset\} \), the weighted structural complexity of an RBAC state is defined as follows:

\[
\text{wsc}(\gamma, W) = w_f \cdot |R| + w_{rec} \cdot |UA| + w_{rel} \cdot |PA| + w_{sh} \cdot |t_r(RH)|
\]

Where

- \( Q^+ \) is the set of all non-negative rational numbers,
- \( |.| \) : the size of the set or relation

A Measure for Minimal Perturbation

The similarity between two roles:

Three different similarity measures:
- Permission centric
- User centric
- Hierarchy relation centric

The role Role Similarity \( \text{sim}(r_1, r_2) \) is defined by combining all these measures with adjustable weights.

The similarity between two role sets \( R_{51} \) and \( R_{52} \):

\[
\forall r_1 \in R_{51}, \exists M_{r_2} \in R_{52}: \text{sim}(r_1, r_2)
\]

\[
\text{if } r_1 \neq r_2 \text{ then } r_1 \neq r_2.
\]

In this step every role in \( R_{51} \) is matched with exactly one distinct role in \( R_{52} \).

There are some roles in \( R_{51} \) that have not been matched with any role from \( R_{52} \).

Define a threshold \( t \) and consider only roles that have a similarity measure above the threshold.

Take average over all of chosen similarities.

**The Proposed Algorithm**

**The Proposed Algorithm**
- First phase- generate the reduced concept lattice
- Second phase- prune this lattice and select the final RBAC state.
- The greedy algorithm prunes the reduced concept lattice based on combination function that defined above.

**Example**

**Results**
- Generates significantly fewer roles than the original state.
- Smaller \( \text{wsc} \) than the HierarchicalMiner and closer to the optimal solution.
- Provides better results compared to VAG algorithm.

**Challenges and Directions**
- Parameterized roles
- Separation of Duty Constraints
- Roles with semantic meanings
- Role mining in multi-domain environments

**Future Work**
- Evaluate the proposed algorithm using real data, and comparing it with the existing approaches.
- Propose an approach that considers separation of duty constrains and its effects on process of migrating to RBAC.