**University & Hospital Policy Test Case**

**1. Basic Test Case**

**Roles:** Professor (Dr\_Li, Dr\_Wang), Doctor (Doctor\_A)

**Resource:** MedicalRecords (StudentA\_Record, StudentB\_Record, StudentC\_Record)

**Action:** Read

**Rules:**

**ProfessorPolicy**

(Dr\_Li, Read, StudentA\_Record)->Deny

(Dr\_Li, Read, StudentB\_Record)->Deny

(Dr\_Li, Read, StudentC\_Record)->Deny

(Dr\_Wang, Read, StudentA\_Record)->Deny

(Dr\_Wang, Read, StudentB\_Record)->Deny

(Dr\_Wang, Read, StudentC\_Record)->Deny

**DoctorPolicy**

(Doctor\_A, Read, StudentA\_Record)->Permit

(Doctor\_A, Read, StudentB\_Record)->Permit

(Doctor\_A, Read, StudentC\_Record)->Permit

**2. Test case with Error Type #1 <access conflict>**

**Roles:** Professor (Dr\_Li, Dr\_Wang), Doctor (Doctor\_A, Dr\_Li (this modification will generate type#1 error, causing conflict for Dr\_Li accessing to StudentA\_Record))

**Resource:** MedicalRecords (StudentA\_Record, StudentB\_Record, StudentC\_Record)

**Action:** Read

**Rules:**

**ProfessorPolicy**

(Dr\_Li, Read, StudentA\_Record)->Deny

(Dr\_Li, Read, StudentB\_Record)->Deny

(Dr\_Li, Read, StudentC\_Record)->Deny

(Dr\_Wang, Read, StudentA\_Record)->Deny

(Dr\_Wang, Read, StudentB\_Record)->Deny

(Dr\_Wang, Read, StudentC\_Record)->Deny

**DoctorPolicy**

(Doctor\_A, Read, StudentA\_Record)->Permit

(Doctor\_A, Read, StudentB\_Record)->Permit

(Doctor\_A, Read, StudentC\_Record)->Permit

(Dr\_Li, Read, StudentA\_Record)->Permit(conflict with the corresponding rule in ProfessorPolicy)

(Dr\_Li, Read, StudentB\_Record)->Deny

(Dr\_Li, Read, StudentC\_Record)->Deny

**Property (used to check the error):**

(Prefessor = Dr\_Li) & (Doctor = Dr\_Li) & (MedicalRecords = StudentA\_Record) & (Action = Read) -> decision = Permit

**Result on ACPT: (NuSMV Verification for Merged Policies)**

(Prefessor = Dr\_Li) & (Doctor = Dr\_Li) & (MedicalRecords = StudentA\_Record) & (Action = Read) -> decision = Permit in ProfessorPolicy is **False**

(Prefessor = Dr\_Li) & (Doctor = Dr\_Li) & (MedicalRecords = StudentA\_Record) & (Action = Read) -> decision = Permit in DoctorPolicy is **True**

**3. Test case with Error Type #2 <no object>**

**Roles:** Professor (Dr\_Li, Dr\_Wang), Doctor (Doctor\_A)

**Resource:** MedicalRecords (StudentA\_Record, StudentB\_Record, StudentC\_Record, StudentD\_Record (this resource is added to generate type#2 error, it is not defined in all rules))

**Action:** Read

**Rules:**

**ProfessorPolicy**

(Dr\_Li, Read, StudentA\_Record)->Deny

(Dr\_Li, Read, StudentB\_Record)->Deny

(Dr\_Li, Read, StudentC\_Record)->Deny

(Dr\_Wang, Read, StudentA\_Record)->Deny

(Dr\_Wang, Read, StudentB\_Record)->Deny

(Dr\_Wang, Read, StudentC\_Record)->Deny

**DoctorPolicy**

(Doctor\_A, Read, StudentA\_Record)->Permit

(Doctor\_A, Read, StudentB\_Record)->Permit

(Doctor\_A, Read, StudentC\_Record)->Permit

**Property (used to check the error):**

(Prefessor = Dr\_Li) & (MedicalRecords = StudentD\_Record) & (Action = Read) -> decision = Permit

**Result on ACPT: (NuSMV Verification for Combined Policies)**

(Prefessor = Dr\_Li) & (MedicalRecords = StudentD\_Record) & (Action = Read) -> decision = Permit is **False**

**4. Test case with Error Type #3 <undecided>**

**Roles:** Professor (Dr\_Li, Dr\_Wang), Doctor (Doctor\_A)

**Resource:** MedicalRecords (StudentA\_Record, StudentB\_Record, StudentC\_Record, StudentD\_Record (this resource is added to generate type#3 error, because every role has been assigned access to it except Dr\_Wang))

**Action:** Read

**Rules:**

**ProfessorPolicy**

(Dr\_Li, Read, StudentA\_Record)->Deny

(Dr\_Li, Read, StudentB\_Record)->Deny

(Dr\_Li, Read, StudentC\_Record)->Deny

(Dr\_Li, Read, StudentD\_Record)->Deny

(Dr\_Wang, Read, StudentA\_Record)->Deny

(Dr\_Wang, Read, StudentB\_Record)->Deny

(Dr\_Wang, Read, StudentC\_Record)->Deny

**DoctorPolicy**

(Doctor\_A, Read, StudentA\_Record)->Permit

(Doctor\_A, Read, StudentB\_Record)->Permit

(Doctor\_A, Read, StudentC\_Record)->Permit

(Doctor\_A, Read, StudentD\_Record)->Permit

**Property (used to check the error):**

(Prefessor = Dr\_Wang) & (MedicalRecords = StudentD\_Record) & (Action = Read) -> decision = Permit

**Result on ACPT: (NuSMV Verification for Combined Policies)**

(Prefessor = Dr\_Wang) & (MedicalRecords = StudentD\_Record) & (Action = Read) -> decision = Permit **is False**