**sig** Validity{

value: **one String**

}{

**all** v: value | (v="VALID" **or** v="EXPIRED")

}

**sig** Card{

validity: **one** Validity

}

**sig** User{

drivingLicenceValidity: **one** Validity,

cardSet: **set** Card

}{

#cardSet>0

}

**sig** SafeArea{

contains: **set** SpecialSafeArea

}

**sig** SpecialSafeArea {}

**sig** Car{

parked: **lone** SafeArea,

}

**sig** Reservation{

user: **one** User,

car: **one** Car

}

**sig** Ride{

reservation: **one** Reservation,

statusOfRide: **one String**

}{

**all** v:statusOfRide | (v="IN USE" **or** v="ONLY RESERVED")

}

**sig** Employee{

workingOnCars: **set** Car

}

**fact** oneSPAInOnlyOneSA{

(**all** spa:SpecialSafeArea | **all** **disj** sa1,sa2:SafeArea | ((spa **in** sa1.contains) **implies** (spa **not in** sa2.contains)) **or** ((spa **in** sa2.contains) **implies** (spa **not in** sa1.contains)))

(**all** spa:SpecialSafeArea | **one** sa:SafeArea | spa **in** sa.contains)

}

**fact** expirationOfCards{

**all** r:Reservation, u:User | **some** c:Card | (r.user=u) **implies** (c **in** u.cardSet **and** c.validity.value="VALID")

}

**fact** expirationOfDrivingLicense{

**no** re:Reservation | re.user.drivingLicenceValidity.value="EXPIRED"

}

**fact** oneRideOneReservation{

**no disj** r1, r2: Ride | r1.reservation=r2.reservation

}

**fact** differentCardsForEachUser{

(**all** c:Card | **all disj** u1,u2:User | ((c **in** u1.cardSet) **implies** (c **not in** u2.cardSet)) **or** ((c **in** u2.cardSet) **implies** (c **in** u1.cardSet)))

(**all** c:Card | **one** u:User | c **in** u.cardSet)

}

fact eachReservationMustHaveADifferentCarAndUser{

(**no disj** r1,r2:Reservation | r1.user=r2.user)

(**no disj** r1,r2:Reservation | r1.car=r2.car)

}

**fact** carStatusCondition{

(**all** c:Car, r:Ride | **some** e:Employee | (c=r.reservation.car)**implies**(#r.reservation.car.parked=0 **implies** (r.statusOfRide="IN USE" **or** (c **in** e.workingOnCars))))

(**all** r:Ride | r.statusOfRide="ONLY RESERVED" **implies** #r.reservation.car.parked=1)

}

**fact** alwaysARideForAReservation{

**all** re:Reservation | **one** ri:Ride | re=ri.reservation

}

**fact** workingCarCantBeReserved{

**all** e:Employee, r:Reservation | **no** c:Car | ((c **in** e.workingOnCars)**and**(c=r.car))

}

**fact** occupationCar{

**all** c:Car, e:Employee | (c **in** e.workingOnCars) **implies** (#c.parked=0)

}

**pred** show{

}

**run** show **for** 3