**OCL Constraints- Case Study 2**

1. **context** RegisteredUser

**inv:**

RegisteredUser -> allInstances() -> **isUnique**(username)

1. **context** Sudoku

**inv:**

Sudoku -> allInstances() -> **isUnique**(id\_sudoku)

1. **context** Sudoku

**inv:**

Sudoku.allInstance() -> **select**(s| s.row\_num <> self.row\_num)->size()=1

1. **context** Sudoku

**inv:**

Sudoku.allInstance() -> **select**(s| s.col\_num <> self.col\_num)->size()=1

1. **context** RowCell

**inv:**

RowCell.allInstance() -> **select**(s| s. correct\_value <> self. correct\_value)->size()=1

1. **context** ColumnCell

**inv:**

ColumnCell.allInstance() -> **select**(s| s. correct\_value <> self. correct\_value)->size()=1

1. **context** Region

**inv:**

Region.allInstance() -> **select**(s| s. correct\_value <> self. correct\_value)->size()=1

1. **context** NewPlayer :: effect()

**post:**

play\_att.oclIsNew() **and** oclIsTypeOf(Player) **and** UserHasAttributes(play\_att)

1. **context** NewAdministrator :: effect()

**post:**

admin.oclIsNew() **and** oclIsTypeOf(Administrator) **and** UserHasAttributes(admin)

1. **context** MailUpdate :: effect()

**post:**

registeredUser.mail = mail

1. **context** PasswordChange :: effect()

**post:**

registeredUser.password = new\_password

1. **context** SudokuChoice :: effect()

**post:**

player. currentSudoku = Sudoku

1. **context** Undo:: effect()

**post:**

Undo. oclIsNew() **and** Undo.oslIsTypeOf(UndoMove) **and**

Undo. nonPredefinedCell = sudoku. lastDisposableMove

1. **context** Redo:: effect()

**post:**

Redo. oclIsNew() **and** Redo.oslIsTypeOf(RedoMove) **and**

Redo. nonPredefinedCell = sudoku. lastUndoneMove

1. **context** NewRegisteredUser:: CorrectMail() : Boolean

**body:**

mail. CorrectMail()

1. **context** MailUpdate:: CorrectMail() : Boolean

**body:**

mail. CorrectMail()

1. **context** SudokuChoice:: UnfinishedSudoku() : Boolean

sudoku.finished = false

1. **context** GameMove:: UnfinishedSudoku() : Boolean

sudoku.finished = false

1. **context** IncorrectCellsCheck:: UnfinishedSudoku() : Boolean

sudoku.finished = false

1. **context** CompoundGameMove:: UnfinishedSudoku() : Boolean

sudoku.finished = false

1. **context** SudokuChoice:: SudokuIsTheCurrentOfPlayer() : Boolean

**body:**

player.currentSudoku = Sudoku **and** sudoku.finished = false

1. **context** IncorrectCellsCheck:: SudokuIsTheCurrentOfPlayer() : Boolean

**body:**

player.currentSudoku = Sudoku **and** sudoku.finished = false

1. **context** CompoundGameMove:: SudokuIsTheCurrentOfPlayer() : Boolean

**body:**

player.currentSudoku = Sudoku **and** sudoku.finished = false

1. **context** PutValueInACell:: CellsPartOfCurrentSudoku() : Boolean

**body:**

player.currentSudoku.nonPredefinedCell -> **includes**(nonPredefinedCell)

1. **context** CellCheck:: CellsFromCurrentSudoku() : Boolean

**body:**

player.currentSudoku.nonPredefinedCell -> **includes**(nonPredefinedCell)

1. **context** ClueMove:: value() : ValueCode

**body:**

nonPredefinedCell.correct\_value

1. **context** SolutionMove:: value() : ValueCode

**body:**

nonPredefinedCell.correct\_value