<u>OCL Constraints- Case Study 2</u>

1. **context** RegisteredUser
   inv:
   RegisteredUser -> allInstances() -> **isUnique**(username)

2. **context** Sudoku
   inv:
   Sudoku -> allInstances() -> **isUnique**(id_sudoku)

3. **context** Sudoku
   inv:
   Sudoku.allInstance() -> **select**(s| s.row_num <> self.row_num)->size()=1

4. **context** Sudoku
   inv:
   Sudoku.allInstance() -> **select**(s| s.col_num <> self.col_num)->size()=1

5. **context** RowCell
   inv:
   RowCell.allInstance() -> **select**(s| s. correct_value <> self. correct_value)->size()=1

6. **context** ColumnCell
   inv:
   ColumnCell.allInstance() -> **select**(s| s. correct_value <> self. correct_value)->size()=1

7. **context** Region
   inv:

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

8. **context** NewPlayer :: effect()
   post:
   play_att.oclIsNew() **and** oclIsTypeOf(Player) **and** UserHasAttributes(play_att)

9. **context** NewAdministrator :: effect()
post:
admin.oclIsNew() **and** oclIsTypeOf(Administrator) **and** UserHasAttributes(admin)

10. **context** MailUpdate :: effect()
post:
registeredUser.mail = mail

11. **context** PasswordChange :: effect()
post:
registeredUser.password = new_password

12. **context** SudokuChoice :: effect()
post:
player. currentSudoku = Sudoku

13. **context** Undo:: effect()
post:
Undo. oclIsNew() **and** Undo.oslIsTypeOf(UndoMove) **and**

Undo. nonPredefinedCell = sudoku. lastDisposableMove

14. **context** Redo:: effect()
post:
Redo. oclIsNew() **and** Redo.oslIsTypeOf(RedoMove) **and**

Redo. nonPredefinedCell = sudoku. lastUndoneMove

15. **context** NewRegisteredUser:: CorrectMail() : Boolean
body:
mail. CorrectMail()

16. **context** MailUpdate:: CorrectMail() : Boolean
body:
mail. CorrectMail()

17. **context** SudokuChoice:: UnfinishedSudoku() : Boolean
sudoku.finished = false

18. **context** GameMove:: UnfinishedSudoku() : Boolean
sudoku.finished = false

19. **context** IncorrectCellsCheck:: UnfinishedSudoku() : Boolean
sudoku.finished = false

20. **context** CompoundGameMove:: UnfinishedSudoku() : Boolean
sudoku.finished = false

21. **context** SudokuChoice:: SudokuIsTheCurrentOfPlayer() : Boolean
body:
player.currentSudoku = Sudoku **and** sudoku.finished = false

22. **context** IncorrectCellsCheck:: SudokuIsTheCurrentOfPlayer() : Boolean
body:
player.currentSudoku = Sudoku **and** sudoku.finished = false

23. **context** CompoundGameMove:: SudokuIsTheCurrentOfPlayer() : Boolean
body:
player.currentSudoku = Sudoku **and** sudoku.finished = false

24. **context** PutValueInACell:: CellsPartOfCurrentSudoku() : Boolean
body:
player.currentSudoku.nonPredefinedCell **->** **includes**(nonPredefinedCell)

25. **context** CellCheck:: CellsFromCurrentSudoku() : Boolean
body:
player.currentSudoku.nonPredefinedCell -> **includes**(nonPredefinedCell)

26. **context** ClueMove:: value() : ValueCode
    body:
        nonPredefinedCell.correct_value

27. **context** SolutionMove:: value() : ValueCode
    body:
        nonPredefinedCell.correct_value