SOEN331: Introduction to Formal Methods for Software Engineering Assignment 2: axiomatic specifications

Instructor: Dr. O. Ormandjieva

March 2, 2017

Domain

Let Universities be a set of objects of type University

Invariant

$\forall u : Universities | uni = (n_{uni}, t_{uni}, p_{uni}) \bullet \forall team : t_{uni} | team = (n_{team}, p_{team}) \bullet p_{team} \subseteq p_{uni}$

3.1 add player to the University

Precondition:

$(p \notin u.players\text{-}pool) \wedge \forall team : u.teams | team = (n_{team}, p_{team}) \bullet p \notin p_{team}$

AddPlayerToUniversity(u, p)

Postcondition:

$(u'.players\text{-}pool = u.players\text{-}pool \cup p) \wedge \forall team : u.teams | team = (n_{team}, p_{team}) \bullet p \notin p_{team}$

3.2 add a team

Precondition:

$\forall team : u.teams | team = (n_{team}, p_{team}) \bullet tn \notin n_{team}$

AddTeam(u, tn)

Postcondition:

$u'.players\text{-}pool = u.players\text{-}pool \wedge \exists team : u'.teams | team = (n_{team}, p_{team}) \bullet (n_{team} = tn) \wedge (p_{team} = \emptyset)$

3.3 add a player to a team

Precondition:

$\exists team : u.teams | team = (n_{team}, p_{team}) \bullet (n_{team} = tn) \wedge (p \in u.players\text{-}pool) \wedge (p \notin p_{team})$

AddPlayerToTeam(u, p, tn)

Postcondition:

$\exists team : u.teams | team = (n_{team}, p_{team}) \bullet (n_{team} = tn \land p \in p_{team}) \land (u'.players\text{-}pool = u.players\text{-}pool)$

3.4 delete a player from a team

Precondition:

$\exists team : u.teams | team = (n_{team}, p_{team}) \bullet (n_{team} = tn) \land (p \in p_{team}) \land (p \in u.players\text{-}pool)$

DeletePlayer-FromTeam(u, p, tn)

Postcondition:

$\exists team : u.teams | team = (n_{team}, p_{team}) \bullet (n_{team} = tn) \land (p \notin p_{team}) \land (u'.players\text{-}pool = u.players\text{-}pool)$

3.5 delete a team

Precondition:

$\exists team : u.teams | team = (n_{team}, p_{team}) \bullet n_{team} = tn$

DeleteTeam(u, tn)

Postcondition:

$\forall team : u.teams | team = (n_{team}, p_{team}) \bullet (n_{team} \neq tn) \land (u'.players\text{-}pool = u.players\text{-}pool)$

3.6 delete a player from university

Precondition:

$p \in u.players\text{-}pool$

DeletePlayerFromUniversity(u,p)

Postcondition:

$\forall team : u.teams | team = (n_{team}, p_{team}) \bullet (p \notin p_{team}) \land (p \notin u'.players\text{-}pool)$

3.7 add game

Precondition:

$\exists u_1, u_2 : Universities | u_1 = (n_{u1}, t_{u1}, p_{u1}) \land u_2 = (n_{u2}, t_{u2}, p_{u2}) \bullet (team1 \in t_{u1}) \land (team2 \in t_{u2}) \land (u_1 \neq u_2)$

AddGames(team1, team2)

Postcondition:

$Games' \neq Games \land \exists game : Game | game = (t_1, t_2) \bullet (t_1 = team1) \land (t_2 = team2) \land (Games' = Games \cup game)$