```
┌──────────────── MODULE SplitLess_group_join_leave ────────────────┐
EXTENDS Naturals

CONSTANTS
    USERS,
    POSSIBLE_REPLICA_IDs,
    ASSIGNED_REPLICA,
    INITIAL_MEMBER

VARIABLES
    replicas, actionCounter

 ──────────────────────────────
 Records
 ──────────────────────────────
Group ≜ [ members : [USERS → Nat],
            invited_members : [USERS → Nat]]

Replica ≜ [id : POSSIBLE_REPLICA_IDs,
            group : Group]


 ──────────────────────────────
 Initialization
 ─────────────────────────────
Init ≜
        ∧ LET InitialGroup ≜
            [members ↦ [u ∈ USERS ↦ 0],
             invited_members ↦ [u ∈ USERS ↦ IF u = INITIAL_MEMBER THEN 1 ELSE 0]]
          IN   ∧ replicas = [rid ∈ POSSIBLE_REPLICA_IDs ↦ [group ↦ InitialGroup]]

        replicas = [rid ∈ POSSIBLE_REPLICA_IDs
                        ↦ [id ↦ rid,
                           group ↦ INITIAL_GROUP] ]
        ∧ actionCounter = 0


 ──────────────────────────────
 Helper Actions
 ─────────────────────────────
IsMember(memberCounter) ≜
  memberCounter%2 = 1

IsMemberContender(memberCounter) ≜
    memberCounter%2 = 1


 ──────────────────────────────
 Group Actions
 ────────────────────────────
InviteMember ≜
    ∃ actor, newMember ∈ USERS :
    ∃ rid ∈ POSSIBLE_REPLICA_IDs :
        ∧ ASSIGNED_REPLICA[actor] = rid
        ∧ IsMember(replicas[rid].group.members[actor])
        ∧ ¬IsMember(replicas[rid].group.members[newMember])
        ∧ ¬IsMemberContender(replicas[rid].group.invited_members[newMember])
        ∧ LET newReplica ≜
            [replicas[rid] EXCEPT !.group.invited_members[newMember] = @ + 1]
            IN
                ∧ replicas′ = [replicas EXCEPT ![rid] = newReplica]
                ∧ actionCounter′ = actionCounter + 1

AcceptInvitation ≜
    ∃ actor ∈ USERS :
    ∃ rid ∈ POSSIBLE_REPLICA_IDs :
        ∧ ASSIGNED_REPLICA[actor] = rid
        ∧ IsMemberContender(replicas[rid].group.invited_members[actor])
        ∧ LET newReplica ≜
            [replicas[rid] EXCEPT
                !.group.invited_members[actor] = @ + 1,
                !.group.members[actor] = @ + 1]
            IN
                ∧ replicas′ = [replicas EXCEPT ![rid] = newReplica]
                ∧ actionCounter′ = actionCounter + 1

LeaveGroup ≜
    ∃ actor ∈ USERS :
    ∃ rid ∈ POSSIBLE_REPLICA_IDs :
        ∧ ASSIGNED_REPLICA[actor] = rid
        ∧ IsMember(replicas[rid].group.members[actor])
        ∧ LET newReplica ≜
            [replicas[rid] EXCEPT !.group.members[actor] = @ + 1]
            IN
                ∧ replicas′ = [replicas EXCEPT ![rid] = newReplica]
                ∧ actionCounter′ = actionCounter + 1


 ──────────────────────────────
 Merge action
 ──────────────────────────────
MergeReplicas ≜
    ∃ ownRid, otherRid ∈ POSSIBLE_REPLICA_IDs :
        ∧ ownRid ≠ otherRid
        ∧ LET
```

```
            merged_members ≜
                [u ∈ USERS ↦
                  CHOOSE n ∈ {replicas[ownRid].group.members[u], replicas[otherRid].group.members[u]} :
                              n ≥ replicas[ownRid].group.members[u] ∧ n ≥ replicas[otherRid].group.members[u]]
            merged_persumed_membrs ≜
                [u ∈ USERS ↦
                  CHOOSE n ∈ {replicas[ownRid].group.invited_members[u], replicas[otherRid].group.invited_mer
                              n ≥ replicas[ownRid].group.invited_members[u] ∧ n ≥ replicas[otherRid].group.invit
            merged_group ≜
                [replicas[ownRid].group EXCEPT !.members = merged_members,
                                               !.invited_members = merged_persumed_membrs]
           IN    ∧ replicas′ = [replicas EXCEPT ![ownRid].group = merged_group]
                 ∧ UNCHANGED actionCounter


 ────────────────────────
 Next relation
 ────────────────────────
Next ≜
    ∨ InviteMember
    ∨ AcceptInvitation
    ∨ LeaveGroup
    ∨ MergeReplicas
    ∨ UNCHANGED ⟨replicas, actionCounter⟩


 ────────────────────────
 Invariants
 ────────────────────────
TypeOK ≜
  ∀ rid ∈ POSSIBLE_REPLICA_IDs :
    ∧ replicas[rid].group
          ∈ Group

 ────────────────────────
 Liveness Helper
 ────────────────────────
AllReplicasHaveAtLeastGroupMemberCounter(user, member_counter) ≜
    ∀ rid ∈ POSSIBLE_REPLICA_IDs :
        ∧ replicas[rid].group.members[user] ≥ member_counter

AllReplicasHaveAtLeastGroupPersumedMemberCounter(user, persumend_member_counter) ≜
    ∀ rid ∈ POSSIBLE_REPLICA_IDs :
        ∧ replicas[rid].group.invited_members[user] ≥ persumend_member_counter

 ────────────────────────
 Liveness
 ──────────────────────────
Liveness_GroupMemberShipPropagates ≜
    ∀ rid ∈ POSSIBLE_REPLICA_IDs :
       ∀ user ∈ USERS :
             □◇AllReplicasHaveAtLeastGroupMemberCounter(user, replicas[rid].group.members[user])

Liveness_PersumedGroupMemberShipPropagates ≜
    ∀ rid ∈ POSSIBLE_REPLICA_IDs :
       ∀ user ∈ USERS :
             □◇AllReplicasHaveAtLeastGroupPersumedMemberCounter(user, replicas[rid].group.invited_mem


Safety Helper
MemberOnlyAfterInvitation ≜
  ∀ rid ∈ POSSIBLE_REPLICA_IDs :
  ∀ u ∈ USERS :
     ∧ ASSIGNED_REPLICA[u] = rid
     ∧ ¬IsMember(replicas[rid].group.members[u])
     ∧ IsMember(replicas′[rid].group.members[u])
     ⇒
       ∧ IsMemberContender(replicas[rid].group.invited_members[u])

NoDecreaseMembershipCounters ≜
  ∀ rid ∈ POSSIBLE_REPLICA_IDs :
  ∀ u ∈ USERS :
       replicas′[rid].group.members[u]
           ≥ replicas[rid].group.members[u]
     ∧ replicas′[rid].group.invited_members[u]
           ≥ replicas[rid].group.invited_members[u]

Safety
Safety_MemberOnlyAfterInvitation ≜
  □[MemberOnlyAfterInvitation]_⟨replicas, actionCounter⟩

Safety_NoDecreaseMembershipCounters ≜
  □[NoDecreaseMembershipCounters]_⟨replicas, actionCounter⟩


 ──────────────────────────
 Specification
 ──────────────────────────
Spec ≜ Init ∧ □[Next]_⟨replicas, actionCounter⟩
FairSpec ≜ Spec ∧ WF_⟨replicas, actionCounter⟩(MergeReplicas)

└──────────────────────────────────────────────────────────────┘
\* Modification History
\* Last modified Thu Jan 15 10:42:22 CET 2026 by floyd
```

\ * Created *Mon Nov* 24 10:30:46 *CET* 2025 by *floyd*