```
THEORY Group
  IMPORT THEORY PROJECTS
    /SimpleDEq THEORIES /SimpleDEq/Monoid.dtf|org.eventb.theory.core.deployedTheoryRoot#Monoid
  TYPE PARAMETERS G
  OPERATORS
    invertible <predicate> (op: (G × G) → G, e: G)
      well−definedness G ≠ ∅
      direct definition
        ∀ x · x ∈ G ⇒ (∃ y · y ∈ G ⇒ (op(x ↦ y) = e ∧ op(y ↦ x) = e))
    commutative <predicate> (op: (G × G) → G)
      well−definedness G ≠ ∅
      direct definition
        ∀ x, y · x ∈ G ∧ y ∈ G ⇒ op(x ↦ y) = op(y ↦ x)
    Group <predicate> (op: (G × G) → G, e: G)
      well−definedness G ≠ ∅
      direct definition
        Monoid(op, e) ∧ invertible(op, e)
    AbelianGroup <predicate> (op: (G × G) → G, e: G)
      well−definedness G ≠ ∅
      direct definition
        Group(op, e) ∧ commutative(op)
    inverses <predicate> (op: (G × G) → G, e: G, x: G, y: G)
      well−definedness G ≠ ∅
      direct definition
        invertible(op, e) ⇒ (op(x ↦ y) = e ∧ op(y ↦ x) = e)
  THEOREMS
    inversesCommutative:
      ∀ op, e, x, y · op ∈ ((G × G) → G) ∧ e ∈ G ∧ x ∈ G ∧ y ∈ G ∧ Group(op, e) ⇒
        (inverses(op, e, x, y) ⇔ inverses(op, e, y, x))
    latinSquare:
      ∀ op, e, x, y · op ∈ ((G × G) → G) ∧ e ∈ G ∧ x ∈ G ∧ y ∈ G ∧ Group(op, e) ⇒ (
        ∃ g · g ∈ G ⇒ (op(x ↦ g) = y ∧ (∀ g2 · g2 ∈ G ∧ op(x ↦ g2) = y ⇒ g = g2))
      )
    leftCancellation:
      ∀ op, e· op ∈ ((G × G) → G) ∧ e ∈ G ∧ Group(op, e) ⇒ (
        ∀ a, b, c · a ∈ G ∧ b ∈ G ∧ c ∈ G ⇒
          ((op(a ↦ b) = op(a ↦ c)) ⇔ (b = c))
      )
    rightCancellation:
      ∀ op, e· op ∈ ((G × G) → G) ∧ e ∈ G ∧ Group(op, e) ⇒ (
        ∀ a, b, c · a ∈ G ∧ b ∈ G ∧ c ∈ G ⇒
          ((op(b ↦ a) = op(c ↦ a)) ⇔ (b = c))
      )
    inverseEqn:
      ∀ op, e· op ∈ ((G × G) → G) ∧ e ∈ G ∧ Group(op, e) ⇒ (
        ∀ x, y · x ∈ G ∧ y ∈ G ⇒ (
          op(x ↦ y) = e ⇔ inverses(op, e, x, y)
        )
      )
    zeroInverse:
      ∀ op, e · op ∈ ((G × G) → G) ∧ e ∈ G ∧ invertible(op, e) ∧ neutral(op, e) ⇒ inverses(op, e, e,
         e)
END
```