Simple properties of Groups in Magma

Meijun Liu

December 5, 2025

Simple Properties of Groups

Simple properties

```
    // Symmetric group S_4
    >G := SymmetricGroup(4);   //input
     IsAbelian(G);
     IsNilpotent(G);
     IsSoluble(G);
     IsSimple(G);
     #G;
     Z := Center(G);
     Z;
     Order(Z);
     >false; false; true; false; 24;
     Permutation group Z acting on a set of cardinality 4
     Order = 1
     1
```

Simple Properties of Groups

Character table

```
>G := SymmetricGroup(4);
ct := CharacterTable(G);
print "Character table of S4:";
print "  Number of irreducible characters:", #ct;
print "  Character degrees:", [Degree(ct[i]) : i in [1..#ct]];
print "  Character table:";
for i := 1 to #ct do
chi := ct[i];
print "    Character", i, ": degree =", Degree(chi), ", values =", [chi(c[3]) : c in
ConjugacyClasses(G)];
end for;
> Character table of S4:
Number of irreducible characters: 5
Character degrees: [ 1, 1, 2, 3, 3 ]
Character table:
Character 1 : degree = 1 , values = [1,1,1,1,1]
Character 2 : degree = 1 , values = [1,1,-1,1,-1]
Character 3 : degree = 2 , values = [2,2,0,-1,0]
Character 4 : degree = 3 , values = [3,-1,-1,0,1]
Character 5 : degree = 3 , values = [3,-1,1,0,-1]
```

Simple Properties of Groups

Subgroups

```
// Symmetric group S_4
>G := SymmetricGroup(4);    //input
 Subgroups(G);
>Conjugacy classes of subgroups     //output
------------------------------
[ 1]    Order 1             Length 1
[ 2]    Order 2             Length 3     (1, 4)(2, 3)
[ 3]    Order 2             Length 6     (3, 4)
[ 4]    Order 3             Length 4     (2, 3, 4)
[ 5]    Order 4             Length 3     (3, 4),(1, 2)(3, 4)
[ 6]    Order 4             Length 1     (1, 4)(2, 3),(1, 3)(2, 4)
[ 7]    Order 4             Length 3     (1, 4, 2, 3),(1, 2)(3, 4)
[ 8]    Order 6             Length 4     (3, 4),(2, 3, 4)
[ 9]    Order 8             Length 3     (3, 4),(1, 4)(2, 3),(1, 3)(2, 4)
[10]    Order 12            Length 1     (2, 3, 4),(1, 4)(2, 3),(1, 4)(2, 3)
[11]    Order 24            Length 1     (3, 4),(2, 3, 4),(1, 4)(2, 3),(1, 3)(2, 4)
```

Simple Properties of Groups

Normal subgroups

```
    // Symmetric group S_4
    >G := SymmetricGroup(4);   //input
    NormalSubgroups(G);
    >Conjugacy classes of subgroups     //output
    ------------------------------

  [1]     Order 1              Length 1
  Permutation group acting on a set of cardinality 4
  Order = 1
  [2]     Order 4              Length 1
  Permutation group acting on a set of cardinality 4
  Order = 4 = 2^2
  (1, 4)(2, 3)
  (1, 3)(2, 4)
  [3]     Order 12             Length 1
  Permutation group acting on a set of cardinality 4
  Order = 12 = 2^2 * 3
  (2, 3, 4)
  (1, 4)(2, 3)
  (1, 3)(2, 4)
  [4]     Order 24             Length 1
  Permutation group acting on a set of cardinality 4
  Order = 24 = 2^3 * 3
  (3, 4)
  (2, 3, 4)
  (1, 4)(2, 3)
  (1, 3)(2, 4)
```

Simple Properties of Groups

Derived Subgroups

```
>G := SymmetricGroup(4);
D := DerivedSubgroup(G);
#D;
Generators(D);
IsIsomorphic(D, AlternatingGroup(4));
DerivedLength(D);
#D eq #DerivedSubgroup(D);
>12;
{(1, 2, 3), (2, 3, 4), (1, 2, 4)};
true Isomorphism of GrpPerm: D, Degree 4, Order 2^2 * 3 into GrpPerm: $, Degree 4, Order 2^2 * 3 induced by
(1, 2, 3) |--> (1, 2, 3)
(2, 3, 4) |--> (2, 3, 4)
(1, 2, 4) |--> (1, 2, 4);
2;
false
```

Simple Properties of Groups

Sylow Subgroups

```
>G := SymmetricGroup(4);
primes := PrimeDivisors(#G);
for p in primes do
    sylow := SylowSubgroup(G, p);
      print "Sylow", p, "-subgroup:";
    print "  Order:", #sylow;
    print "  Generators:", Generators(sylow);
    print "  Is normal:", IsNormal(G, sylow);
    print "  Is cyclic:", IsCyclic(sylow);
    print "  Is abelian:", IsAbelian(sylow);
    print "";
    end for;
>Sylow 2-subgroup:
Order: 8
Generators: [ (1, 2, 3, 4), (1, 3) ]
Is normal: false
Is cyclic: false
Is abelian: false

Sylow 3-subgroup:
Order: 3
Generators: [ (1, 2, 3) ]
Is normal: false
Is cyclic: true
Is abelian: true
```

Simple Properties of Groups

Group action–natural set {1234}

```
>S4 := SymmetricGroup(4);
X := {1,2,3,4};
GSetX := GSet(S4, X);  //define the action of S_4 on the set X
print "  Set X =", X;
print "  Set size |X| =", #X;
print "";
orbits := Orbits(S4, GSetX);  // Compute orbits
print "    Number of orbits:", #orbits;
for i in [1..#orbits] do
  orbit := orbits[i];
  print "    Orbit", i, ": size =", #orbit, ", representative =", Representative(orbit);
end for;
print "";
print "  Stabilizer analysis:";// Stabilizer analysis
for point in X do
stab := Stabilizer(S4, point);
print "    Stabilizer of point", point, ":";
print "      Order:", #stab;
print "      Isomorphic to S3?", IsIsomorphic(stab, SymmetricGroup(3));
orbit_size := #Orbit(S4, point);  // Verify the Orbit-Stabilizer theorem
print "      Orbit-Stabilizer theorem check:", orbit_size * #stab, "=", #S4,
"?", orbit_size * #stab eq #S4;
end for;
```

Simple Properties of Groups

Group action–natural set $\{1234\}$

```
 >Set X = { 1, 2, 3, 4 }
Set size |X| = 4

Orbit analysis:
Number of orbits: 1
Orbit 1: size = 4, representative = 1

Stabilizer analysis:
Stabilizer of point 1:
Order: 6
Isomorphic to S3? true
Orbit-Stabilizer theorem check: 4 * 6 = 24 ? true
Stabilizer of point 2:
Order: 6
Isomorphic to S3? true
Orbit-Stabilizer theorem check: 4 * 6 = 24 ? true
Stabilizer of point 3:
Order: 6
Isomorphic to S3? true
Orbit-Stabilizer theorem check: 4 * 6 = 24 ? true
Stabilizer of point 4:
Order: 6
Isomorphic to S3? true
Orbit-Stabilizer theorem check: 4 * 6 = 24 ? true
```

Simple Properties of Groups

Group action – conjugacy action

```
>S4 := SymmetricGroup(4);
print "  Set: S4 itself, with", #S4, "elements";
print "  Action: conjugation (g\cdot x = g*x*g^(-1))";
print "";
print "  Conjugacy class analysis:";
print "    Number of conjugacy classes:", NumberOfClasses(S4);
classes := Classes(S4);
for i in [1..#classes] do
class := classes[i];
print "    Conjugacy class", i, ":";
print "      Size =", class[2];   // The second element is the size of the class.
print "      Representative:", class[3]; // The third element is the representative.
print "      Cycle type:", CycleStructure(class[3]);
end for;
print "";
```

Simple Properties of Groups

Group action – conjugacy action

```
> Set: S4 itself, with 24 elements
Action: conjugation (g.x = g*x*g^(-1))

Conjugacy class analysis:
Number of conjugacy classes: 5
Conjugacy class 1:
Size = 1
Representative: Id(S4)
Cycle type: [ <1, 4> ]
Conjugacy class 2:
Size = 6
Representative: (1, 2)
Cycle type: [ <1, 2>, <2, 1> ]
Conjugacy class 3:
Size = 3
Representative: (1, 2)(3, 4)
Cycle type: [ <2, 2> ]
Conjugacy class 4:
Size = 8
Representative: (1, 2, 3)
Cycle type: [ <1, 1>, <3, 1> ]
Conjugacy class 5:
Size = 6
Representative: (1, 2, 3, 4)
Cycle type: [ <4, 1> ]
```

Simple Properties of Groups

Group homomorphisms – Sign homomorphism

```
>G := SymmetricGroup(4);
C2 := CyclicGroup(2); // Define the sign homomorphism: maps a permutation to its sign (+-1)
sign_map := hom<G -> C2  x :-> (Sign(x) eq 1) select C2!1 else C2.1>; // Sign(x) returns 1 for even permutations, -1 for odd permutations
kernel_order := #Kernel(sign_map);
print "  Kernel order:", kernel_order;
kernel_is_A4 := IsIsomorphic(Kernel(sign_map), AlternatingGroup(4));// Verify if the kernel is isomorphic to alternating group A4
print "  Kernel is A4:", kernel_is_A4;
image_order := #Image(sign_map);
print "  Image order:", image_order;
is_surjective := #Image(sign_map) eq #C2; // Verify if it's surjective (image equals the whole C2)
print "  Is surjective:", is_surjective;
>Kernel order: 12
Kernel is A4: true
Image order: 2
Is surjective: true
```

Simple Properties of Groups

Group homomorphisms – Natural quotient map

```
>G := SymmetricGroup(4);
klein := sub<G | (1,2)(3,4), (1,3)(2,4)>;
Q, phi := quo<G | klein>;
print "  Quotient order:", #Q;
print "  Quotient isomorphic to S3:", IsIsomorphic(Q, SymmetricGroup(3));
print "Composition series:";  // Composition series
CS := CompositionSeries(G);
for i := 1 to #CS do
  H := CS[i];
  print "  Subgroup of order:", #H;
end for;
>Quotient order: 6
Quotient isomorphic to S3: true Isomorphism of GrpPerm: Q, Degree 3, Order 2 *
3 into GrpPerm: $, Degree 3, Order 2 * 3 induced by
(2, 3) |--> (2, 3)
(1, 2) |--> (1, 2)

Composition series:
Subgroup of order: 24
Subgroup of order: 12
Subgroup of order: 4
Subgroup of order: 2
Subgroup of order: 1
```

Simple Properties of Groups

Direct Product

```
>S4 := SymmetricGroup(4);
C2 := CyclicGroup(2);
AutS4 := AutomorphismGroup(S4);
phi_trivial := hom<C2 -> AutS4 | x :-> Id(AutS4)>;
G := SemidirectProduct(S4, C2, phi_trivial);
#G;
G;
>48;
Permutation group G acting on a set of cardinality 6
Order = 48 = 2^4 * 3
(1, 2, 3, 4)
(1, 2)
(5, 6)
```

Simple Properties of Groups

Semi-direct Product

```
>S4 := SymmetricGroup(4);
C2 := CyclicGroup(2);
AutS4 := AutomorphismGroup(S4);
phi_trivial := hom<C2 -> AutS4 | x :-> Id(AutS4)>;
G := SemidirectProduct(S4, C2, phi_trivial);
#G;
G;
>48;
Permutation group G acting on a set of cardinality 6
Order = 48 = 2^4 * 3
(1, 2, 3, 4)
(1, 2)
(5, 6)
G := SemidirectProduct(N, H, phi);
```

Semi-direct Product

```
        >S3 := SymmetricGroup(3);
        C2 := CyclicGroup(2);
        aut := hom<S3 -> S3 | x :-> (S3!(1,2)) * x * (S3!(1,2))>;
        phi := hom<C2 -> AutomorphismGroup(S3) | C2.1 :-> aut>;
        K := SemidirectProduct(S3, C2, phi); //G := SemidirectProduct(N, H, phi), N is normal
        subgroup, H is subgroup, phi is a homomorphism from H to Aut(N).
        #K;
        > 12
G := SemidirectProduct(N, H, phi);
```

Thank You!