

WeylGroups package

A subpackage for QuantumGroups v2.
Version 2.0, June 18, 2005, Scott Morrison

Introduction

This package calculates Weyl groups, and provides decompositions of the longest word as a product of simple reflections.

Implementation

```
BeginPackage["QuantumGroups`WeylGroups`",
  {"QuantumGroups`", "QuantumGroups`RootSystems`"}];
```

```
PositiveRoots::usage = "";
```

```
WeylGroup::usage =
  "WeylGroup[Γ] returns a list of matrices, representing the Weyl group
  elements in the fundamental weight basis.;"
```

```
LongestWordDecomposition::usage =
  "LongestWordDecomposition[Γ] returns the lexicographically
  smallest decomposition of the longest element of the Weyl group.;"
```

```
LongestWord::usage = "LongestWord[Γ] returns the longest
  element of the Weyl group, in the fundamental weight basis.;"
```

```
Begin["`Private`"];
```

```
PositiveRoots[Γ_] :=
  PositiveRoots[Γ] = With[{l = LongestWordDecomposition[Γ], s = SimpleRoots[Γ]},
    Table[
      Fold[SimpleReflection[Γ, #2][#1] &,
        s[[1[[i]]], Reverse[Take[l, i - 1]]], {i, 1, Length[l]}]]
```

```
WeylReflectionMatrix[Γ_, i_] := WeylReflectionMatrix[Γ, i] =
  Transpose[SimpleReflection[Γ, i] /@ IdentityMatrix[Rank[Γ]]]
```

```

WeylGroup[T_] := WeylGroup[T] =
Module[{indexedSimpleReflections = Table[{{i}, WeylReflectionMatrix[T, i]},
  {i, 1, Rank[T]}], newElements, allElements = {}},
newElements = {{{}, IdentityMatrix[Rank[T]]}};
While[newElements != {},
  allElements = allElements ~Join~ newElements;
  newElements = Flatten[Outer[{#1[[1]] ~Join~ #2[[1]], #1[[2]].#2[[2]]} &,
    indexedSimpleReflections, newElements, 1, 1];
  newElements = Union[Complement[newElements, allElements],
    SameTest -> (#1[[2]] == #2[[2]] &)], SameTest -> (#1[[2]] == #2[[2]] &)];
];
LongestWord[T] = Last[allElements][[2]];
LongestWordDecomposition[T] = Last[allElements][[1]];
Transpose[allElements][[2]]
]

```

```

LongestWord[T_] := (Dot @@ (WeylReflectionMatrix[T, #] & /@ LongestWordDecomposition[T]))

```

```

LongestWordDecomposition[T_] := (WeylGroup[T];
  LongestWordDecomposition[T])

```

```

LongestWordDecomposition[A_n] :=
  LongestWordDecomposition[A_n] = Flatten[Table[Reverse[Range[j]], {j, 1, n}]]

```

```

LongestWordDecomposition[T: (B | C)_n] := LongestWordDecomposition[T] = Flatten[
  Table[Take[Reverse[Range[n]], {Max[1, k], Min[k + n - 1, n]}], {k, n, -n + 2, -1}]]

```

```

LongestWordDecomposition[D_n] :=
  LongestWordDecomposition[D_n] = Flatten[Table[Reverse[Range[j]], {j, 1, n - 2}] ~Join~
  {n - 1} ~Join~ Reverse[Range[n - 2]] ~Join~ {n} ~Join~ Reverse[Range[n - 2]] ~
  Join~ Table[{n - Mod[j, 2]} ~Join~ Range[n - 2, j + 1, -1], {j, 1, n - 2}]]

```

```

LongestWordDecomposition[E_6] = {1, 2, 3, 1, 4, 2, 3, 1, 4, 3, 5,
  4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1};

```

```

LongestWordDecomposition[E_7] =
  {1, 2, 3, 1, 4, 2, 3, 1, 4, 3, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6,
  5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 4, 5, 6, 7};

```

```

LongestWordDecomposition[E_8] =
  {1, 2, 3, 1, 4, 2, 3, 1, 4, 3, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2,
  6, 5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 4, 5,
  6, 7, 8, 7, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 4, 5, 6, 7,
  8, 7, 6, 5, 4, 2, 3, 1, 4, 3, 5, 4, 2, 6, 5, 4, 3, 1, 7, 6, 5, 4, 2, 3, 4, 5, 6, 7, 8};

```

```
LongestWordDecomposition[F4] =  
  {1, 2, 1, 3, 2, 1, 3, 2, 3, 4, 3, 2, 1, 3, 2, 3, 4, 3, 2, 1, 3, 2, 3, 4};
```

```
LongestWordDecomposition[G2] = {1, 2, 1, 2, 1, 2};
```

```
End[];
```

```
EndPackage[];
```