pdf

Pensieve header: The WG algebra in the most basic form.

pdf

In[•]:=
```
Once[<< KnotTheory`];
HL[ℰ_] := Style[ℰ, Background → If[TrueQ@ℰ, , ]];
χ[cond_] := If[TrueQ[cond], 1, 0];
```

pdf

```
Loading KnotTheory` version of February 2, 2020, 10:53:45.2097.
Read more at http://katlas.org/wiki/KnotTheory.
```

pdf

In[•]:=
```
DeclareGroup[G_Symbol, S_n_] := Module[{α, β, e},
  G /: Ord[G] = n!;
  G /: Elements[G] = PermutationCycles /@ (Permutations@Range@n);
  Do[G /: g[G, α] = e = Elements[G][[α]]; G /: ind[G, e] = α,
   {α, Ord[G]}];
  Do[G /: m[G, α, β] = ind[G, g[G, α] ~PermutationProduct~ g[G, β]],
   {α, Ord[G]}, {β, Ord[G]}];
  Do[G /: inv[G, α] = ind[G, InversePermutation[g[G, α]]],
   {α, Ord[G]}]]
```

pdf

In[•]:=
```
g[α_] := g[$G, α]; ind[α_] := ind[$G, α];
m[α_, β_] := m[$G, α, β]; inv[α_] := inv[$G, α];
```

pdf

In[•]:=
```
m_{i_,j_→k_}[ℰ_] := Expand[ℰ] /. W_i[α_, β_] W_j[γ_, δ_] :↦ χ[m[α, β] == m[β, γ]] W_k[α, m[β, δ]]
```

pdf

In[•]:=
```
R_{i_,j_} := Sum[W_i[α, 1] W_j[β, α], {α, Ord[$G]}, {β, Ord[$G]}];
R̄_{i_,j_} := Sum[W_i[α, 1] W_j[β, inv@α], {α, Ord[$G]}, {β, Ord[$G]}];
η_{i_} := Sum[W_i[α, 1], {α, Ord[$G]}];
```

pdf

In[•]:=
```
DeclareGroup[$G = S3, S_3];
Table[m[i, j], {i, Ord[S3]}, {j, Ord[S3]}] // MatrixForm
```

Out[•]//MatrixForm=
pdf

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 4 & 3 & 6 & 5 \\ 3 & 5 & 1 & 6 & 2 & 4 \\ 4 & 6 & 2 & 5 & 1 & 3 \\ 5 & 3 & 6 & 1 & 4 & 2 \\ 6 & 4 & 5 & 2 & 3 & 1 \end{pmatrix}$$

pdf

```
In[•]:= Short[lhs = R1,2 R4,3 R5,6 // m1,4→1 // m2,5→2 // m3,6→3]
        rhs = R2,3 R1,4 R5,6 // m1,5→1 // m2,6→2 // m3,4→3;
        HL[lhs == rhs]
```

pdf

```
Out[•]//Short= W1[1, 1] W2[1, 1] W3[1, 1] + W1[2, 1] W2[2, 2] W3[1, 1] +
                 <<212>> + W1[3, 1] W2[5, 3] W3[6, 6] + W1[1, 1] W2[6, 1] W3[6, 6]
```

pdf

```
Out[•]= True
```

pdf

```
In[•]:= Short[lhs = R1,2 R̄3,4 // m1,3→1 // m2,4→2];
        rhs = η1 η2 // Expand;
        HL[lhs == rhs]
```

pdf

```
Out[•]= True
```

pdf

```
In[•]:= Short[lhs = R1,2 R̄3,4 // m1,3→1 // m4,2→2];
        rhs = η1 η2 // Expand;
        HL[lhs == rhs]
```

pdf

```
Out[•]= True
```

pdf

```
In[•]:= Z[K_] := Module[{z},
          z = Expand[Times @@ PD[K] /. x : X[i_, j_, k_, l_] :> If[PositiveQ@x, Rl,i, R̄j,i]];
          Do[z = z // mc⟦1⟧,c⟦j⟧→c⟦1⟧, {c, Skeleton[K]}, {j, 2, Length@c}];
          z]
```

pdf

```
In[•]:= Elements[$G]
```

pdf

```
Out[•]= {Cycles[{}], Cycles[{{2, 3}}], Cycles[{{1, 2}}],
         Cycles[{{1, 2, 3}}], Cycles[{{1, 3, 2}}], Cycles[{{1, 3}}]}
```

pdf

```
        tab = Table[Echo@Timing[K → Z[K]], {K, AllKnots[{3, 5}]}];
```

pdf

```
» {0.578125, Knot[3, 1] → W1[1, 1] + 3 W1[2, 2] + 3 W1[3, 3] + W1[4, 1] + W1[5, 1] + 3 W1[6, 6]}
```

pdf

```
» {24.1563, Knot[4, 1] → W1[1, 1] + W1[2, 1] + W1[3, 1] + W1[4, 1] + W1[5, 1] + W1[6, 1]}
```

pdf

```
» {1174.64, Knot[5, 1] → W1[1, 1] + W1[2, 2] + W1[3, 3] + W1[4, 4] + W1[5, 5] + W1[6, 6]}
```

pdf

```
» {1113.59, Knot[5, 2] → W1[1, 1] + W1[2, 2] + W1[3, 3] + W1[4, 4] + W1[5, 5] + W1[6, 6]}
```

pdf

To do: Tangle algorithm, conjugacy classes, implement and verify Δ etc.