

Pensieve header: The OU Programs (minus counting by gens and rels).

```
In[*]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\OU"];
<< KnotTheory`
```

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

pdf

```
In[*]:= SetAttributes[VD, Orderless]
```

```
In[*]:= bp[i_, j_] :=  $\sigma_{i,j}$ ; bm[i_, j_] :=  $\bar{\sigma}_{i,j}$ ;
```

pdf

```
In[*]:= Tidy[vd_VD] := Module[{ps = Union@@(List@@@vd)},
  Replace[vd, Thread[ps -> Range@Length@ps], {2}]]
```

pdf

```
In[*]:= R12Reduce1[vd_VD] := Tidy@Module[{R2s, R2}, Which[
  Length[R2s = Cases[vd, X_s_[i_, j_] -> X_s_[i + 1, j + 1]]  $\cap$  (List@@vd)] > 0,
  Complement[vd, VD[R2 = First@R2s, R2 /. X_s_[i_, j_] -> X_s_[i - 1, j - 1]]],
  Length[R2s = Cases[vd, X_s_[i_, j_] -> X_s_[i + 1, j - 1]]  $\cap$  (List@@vd)] > 0,
  Complement[vd, VD[R2 = First@R2s, R2 /. X_s_[i_, j_] -> X_s_[i - 1, j + 1]]],
  True, DeleteCases[vd, X_[i_, j_] /; Abs[i - j] == 1]]]
```

pdf

```
In[*]:= R12Reduce[vd_VD] := FixedPoint[R12Reduce1, vd]
```

pdf

```
In[*]:=  $\gamma$ [vd_VD] := Module[{js, s1, i1, j1, s2, i2, j2},
  js = Cases[vd, X_[_, j_] -> j]  $\cap$  Cases[vd, X_[i_, _] -> i - 1];
  If[Length[js] == 0, vd,
  j1 = RandomChoice[js]; i2 = j1 + 1;
  Cases[vd, X_s_[i_, j1] -> (s1 = s; i1 = i)];
  Cases[vd, X_s_[i2, j_] -> (s2 = s; j2 = j)];
  Tidy@Join[Complement[vd, VD[X_s1[i1, j1], X_s2[i2, j2]]],
  VD[X_s2[j1, j2], X_s1[i1, i2], X_s1s2[i1 - s1/3, j2 + s2/3], X_s1s2[i1 + s1/3, j2 - s2/3]]]
  ]]
```

pdf

```
In[*]:=  $\Gamma$ [vd_VD] := FixedPoint[ $\gamma$ , vd, 216]
```

pdf

```
In[*]:=  $\Gamma$ [T_] /; Head[T] != VD :=  $\Gamma$ [VD[T]]
```

pdf

```
 $\bar{\Gamma}$ [vd_VD] := FixedPoint[ $\gamma$ @*R12Reduce, vd, 216]
```

pdf

```
In[ ]:=  $\overline{\Gamma}[T\_]$  /; Head[T] != VD :=  $\overline{\Gamma}[VD[T]]$ 
```

```
In[ ]:= BR[3, {2, -1, -1, -1, -1}] //  $\overline{\Gamma}$ 
```

```
Out[ ]:= VD[EOS[11], EOS[21], EOS[35], X-1[1, 31], X-1[2, 7], X-1[3, 27], X-1[12, 32],
X-1[13, 8], X-1[14, 28], X-1[15, 4], X-1[22, 33], X-1[23, 9], X-1[24, 29],
X-1[25, 5], X1[16, 26], X1[17, 6], X1[18, 30], X1[19, 10], X1[20, 34]]
```

pdf

```
In[ ]:= VPB[n_, {σ---}] := VPB[n, σ];
```

pdf

```
In[ ]:= VD /: vd1_VD ** vd2_VD := Module[{es1, es2, m2},
  es1 = Cases[vd1, EOS[i_] => i];
  m2 = Max[es2 = Cases[vd2, EOS[i_] => i]];
  Tidy[vd1 ∪ Replace[DeleteCases[vd2, _EOS],
    i_ => i/m2 - 1 + es1[[1 + Count[es2, e_ /; i > e]]], {2}]]
]
```

pdf

```
In[ ]:= VD[VPB[n_]] := VD @@ (EOS /@ Range[n]);
VD[VPB[n_, σi,j]] := Tidy@Append[VD @@ (EOS /@ Range[n]), X+1[i - 0.5, j - 0.5]];
VD[VPB[n_,  $\overline{\sigma}_{i,j}$ ]] := Tidy@Append[VD @@ (EOS /@ Range[n]), X-1[i - 0.5, j - 0.5]];
VD[VPB[n_, σ, σ---]] := VD[VPB[n, σ]] ** VD[VPB[n, σ---]]
```

pdf

```
In[ ]:= VPBGenerators[n_] :=
  VPBGenerators[n] = Flatten@Table[{σi,j,  $\overline{\sigma}_{i,j}$ }, {i, n}, {j, DeleteCases[Range@n, i]}];
```

pdf

```
In[ ]:= ProudFollowers[n_, σi,j,  $\overline{\sigma}_{i,j}$ ] := ProudFollowers[n, σi,j] = Module[{p, q, s},
  Flatten@{σi,j, σj,i,  $\overline{\sigma}_{j,i}$ ,
    Table[{σp,q, σq,p,  $\overline{\sigma}_{p,q}$ ,  $\overline{\sigma}_{q,p}$ }, {p, {i, j}}, {q, Complement[Range[n], {i, j}]}],
    Table[{σp,q,  $\overline{\sigma}_{p,q}$ },
      {p, Complement[Range[i + 1, n], {j}]}], {q, Complement[Range[n], {i, j, p}]}]
  }];
ProudFollowers[n_,  $\overline{\sigma}_{i,j}$ ] := ProudFollowers[n,  $\overline{\sigma}_{i,j}$ ] = ProudFollowers[n, σi,j] /. σi,j →  $\overline{\sigma}_{i,j}$ 
```

```
In[ ]:= ProudFollowers[n_, i_Integer] :=
  DeleteCases[Range[Max[Abs[i] - 1, 1], n - 1] ∪ (-Range[Max[Abs[i] - 1, 1], n - 1]), -i];
```

```
In[ ]:= ProudFollowers[5, 3]
```

```
Out[ ]:= {-4, -2, 2, 3, 4}
```

pdf

```
In[*]:= ProudVPBs[n_, 0] := {VPB[n]};
ProudVPBs[n_, 1] := VPB[n, #] & /@ VPBGenerators[n];
ProudVPBs[n_, m_] /; m > 1 := Flatten[
  ProudVPBs[n, m - 1] /. VPB[n, σ___, σ_] => (VPB[n, σ, σ, #] & /@ ProudFollowers[n, σ])]
```

pdf

```
In[*]:= CountOUForms[n_, m_] := Module[{k},
  Length@Union@Flatten@Table[Γ@vpb, {k, 0, m}, {vpb, ProudVPBs[n, k]}]]
```

```
In[*]:= ProudBs[n_, 0] := {BR[n, {}]};
ProudBs[n_, 1] := BR[n, {#}] & /@ (Range[n - 1] ∪ (-Range[n - 1]));
ProudBs[n_, m_] /; m > 1 := Flatten[
  ProudBs[n, m - 1] /. BR[n, {σ___, σ}] => (BR[n, {σ, σ, #}] & /@ ProudFollowers[n, σ])]
```

```
In[*]:= ProudBs[3, 3]
```

```
Out[*]:= {BR[3, {-2, -2, -2}], BR[3, {-2, -2, -1}], BR[3, {-2, -2, 1}], BR[3, {-2, -1, -2}],
BR[3, {-2, -1, -1}], BR[3, {-2, -1, 2}], BR[3, {-2, 1, -2}], BR[3, {-2, 1, 1}],
BR[3, {-2, 1, 2}], BR[3, {-1, -2, -2}], BR[3, {-1, -2, -1}], BR[3, {-1, -2, 1}],
BR[3, {-1, -1, -2}], BR[3, {-1, -1, -1}], BR[3, {-1, -1, 2}], BR[3, {-1, 2, -1}],
BR[3, {-1, 2, 1}], BR[3, {-1, 2, 2}], BR[3, {1, -2, -2}], BR[3, {1, -2, -1}],
BR[3, {1, -2, 1}], BR[3, {1, 1, -2}], BR[3, {1, 1, 1}], BR[3, {1, 1, 2}],
BR[3, {1, 2, -1}], BR[3, {1, 2, 1}], BR[3, {1, 2, 2}], BR[3, {2, -1, -2}],
BR[3, {2, -1, -1}], BR[3, {2, -1, 2}], BR[3, {2, 1, -2}], BR[3, {2, 1, 1}],
BR[3, {2, 1, 2}], BR[3, {2, 2, -1}], BR[3, {2, 2, 1}], BR[3, {2, 2, 2}]}
```

```
In[*]:= CountBs[n_, m_] := Module[{k},
  Length@Union@Flatten@Table[Γ@b, {k, 0, m}, {b, ProudBs[n, k]}]]
```

```
In[*]:= CountBs[3, 3]
```

```
Out[*]:= 47
```

```
In[*]:= AllBs[n_, m_] := DeleteDuplicatesBy[Γ]@Flatten@Table[b, {k, 0, m}, {b, ProudBs[n, k]}]
```

```
In[*]:= Length /@ Table[ProudBs[3, k], {k, 0, 3}]
```

```
Out[*]:= {1, 4, 12, 36}
```

```
In[*]:= AllOUs[n_, 0] := {VD@Flatten@Table[{BT[2 i - 1], EOS[2 i]}, {i, n}]}];
AllOUs[n_, m_] /; m > 0 :=
Sort@Flatten[AllOUs[n, m - 1] /. vd_VD => Module[{BTs, EOSs, k, max0, s},
  BTs = Sort@Cases[vd, BT[i_] => i];
  EOSs = Sort@Cases[vd, EOS[i_] => i];
  max0 = Max[1, Max[Cases[vd, X_[i_, _] => i]]];
  Table[
    Tidy[Append[vd, Xs[p - 0.5, q + 0.5]]],
    {s, {-1, 1}}, {k, Length[BTs]},
    {q, BTs[[k]], EOSs[[k] - 1], {p, Select[BTs, (# >= max0) &]}
  ]]]
```

```
In[*]:= AllROUs[n_, m_] :=
Select[AllOUs[n, m] /. vd_VD => Tidy@DeleteCases[vd, _BT], (# === R12Reduce[#]) &]
```

```
In[*]:=  $\xi$ [vd_VD] := Count[ $\bar{\Gamma}$ [vd], X[_ , _]]
```

```
In[*]:= VD /: ( $\sigma_{i,j}$  | vd_VD) := Switch[Order[ $\xi$ [vd],  $\xi$ [VD[VPB[Count[vd, _EOS],  $\bar{\sigma}_{i,j}$ ]] ** vd]],
  0, Print["OMG, Trouble!"],
  1, False, -1, True];
VD /: ( $\bar{\sigma}_{i,j}$  | vd_VD) := Switch[Order[ $\xi$ [vd],  $\xi$ [VD[VPB[Count[vd, _EOS],  $\sigma_{i,j}$ ]] ** vd]],
  0, Print["OMG, Trouble!"],
  1, False, -1, True];
```

```
In[*]:= VD /: Divisors[vd_VD] := Select[VPBGenerators[Count[vd, _EOS]], (# | vd) &];
VD /: Quotients[vd_VD] :=
 $\bar{\Gamma}$ [VD[VPB[Count[vd, _EOS], # / . { $\sigma \rightarrow \bar{\sigma}$ ,  $\bar{\sigma} \rightarrow \sigma$ }] ** vd] & /@ Divisors[vd];
```

```
In[*]:= OUGraph[n_, m_] := Module[{gens, OUs, k, d, g, q, m1, m2},
  gens = VPBGenerators[n];
  OUs = Flatten@Table[AllROUs[n, k], {k, 0, m}];
  OURule = Dispatch@Thread[OUs -> Range@Length@OUs];
  Graph[
    Range@Length@OUs,
    Union@Flatten@Table[
      m1 = Count[d, X[_ , _]];
      m2 = Count[q =  $\bar{\Gamma}$ [VD[VPB[n, g]] ** d], X[_ , _]];
      If[m2 < m1, Labeled[(d -> q) /. OURule, g], Nothing],
      {d, OUs}, {g, gens}
    ]
  ]
]
```

In[*]:=

```

ExtractVPB[vd_VD] := Module[{n, ds, d},
  n = Count[vd, _EOS];
  If[Length[ds = Divisors[vd]] == 0, VPB[n],
    d = First@Sort[ds];
    q =  $\overline{\Gamma}$ [VD[VPB[n, d /. { $\sigma \rightarrow \overline{\sigma}$ ,  $\overline{\sigma} \rightarrow \sigma$ }}]] ** vd];
    Insert[ExtractVPB[q], d, 2]
  ]];
CF[vpb_VPB] := ExtractVPB[ $\overline{\Gamma}$ [vpb]];

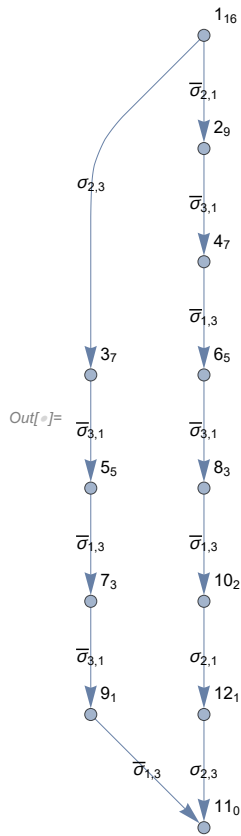
```

```

ExtractionGraph[o, opts___] := Module[{vd, n, gs, vs, es, p, m1, m2, g, q, k},
  gs = VPBGenerators[n = Count[vd =  $\overline{\Gamma}$ [o], _EOS]];
  vs = {vd}; es = {}; p = 0;
  While[p < Length[vs],
    m1 = Count[vd = vs[ $[[++p]]$ ], X[_[_], _]];
    Do[
      m2 = Count[q =  $\overline{\Gamma}$ [VD[VPB[n, g /. { $\sigma \rightarrow \overline{\sigma}$ ,  $\overline{\sigma} \rightarrow \sigma$ }}]] ** vd], X[_[_], _]];
      If[m2 < m1,
        If[! MemberQ[vs, q], AppendTo[vs, q]];
        k = Position[vs, q][[1, 1]];
        AppendTo[es, Labeled[p  $\leftrightarrow$  k, g]]
      ],
      {g, gs}
    ]
  ];
  Graph[Table[Labeled[k, k $_{\text{Length}[vs[[k]]-n}$ ], {k, p}], es, opts]
]

```

In[]:= BR[3, {2, -1, -1, -1, -1}] // ExtractionGraph



```
In[ ]:= VPB[BR[n_, is_List]] := VPB[n, Module[{pi, i},
  pi = Range[n];
  Sequence @@ Table[
    If[i > 0,
      pi[[{i, i + 1}]] = pi[[{i + 1, i}]]; sigma_{pi[[i+1]], pi[[i]}],
      pi[[{-i, -i + 1}]] = pi[[{-i + 1, -i}]]; sigma_{pi[[i], pi[[i+1]]}
    ],
    {i, is}
  ] ]];
VD[br_BR] := VD[VPB@br]
```

```
In[ ]:= RandomBraid[n_, m_] := BR[n, Table[RandomChoice[Range[n - 1] U (-Range[n - 1])], {m}]]
```