

Pensieve Header: The "Cut and Cap" part of the GPV algorithm; continued in Pensieve/Projects/GPV.

<< **KnotTheory`**

Loading KnotTheory` version of April 20, 2009, 14:18:34.482.

Read more at <http://katlas.org/wiki/KnotTheory>.

```

gc = GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[6, 8, 1], Ar[3, 7, -1]]
GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[6, 8, 1], Ar[3, 7, -1]]

Place[{Ar}, {i_, j_}] :=
  {GC[Ar[i, j, 1]], GC[Ar[j, i, 1]], GC[Ar[i, j, -1]], GC[Ar[j, i, -1]]};
Place[{Ar, objs_}, {i_, rest_}] := Flatten[Table[
  Outer[Join,
    Place[{Ar}, {i, {rest}[[k]]}],
    Place[{objs}, Delete[{rest}, k]]
  ],
  {k, Length[{rest]}]
]];

Diagrams[k_.*Ar] := NormalForm[
  Place[Table[Ar, {k}], Range[2k]]
];

NormalForm[gc_GC] := Module[{l, r, a, b},
  Sort[
    l = Length[r = Sort[Flatten[Cases[gc, Ar[a_, b_, _] => {a, b}]]]];
    r = Thread[r -> Range[l]];
    gc /. Ar[a_, b_, s_] => Ar[a /. r, b /. r, s],
    (Min[#1[[1]], #1[[2]]] < Min[#2[[1]], #2[[2]]) &
  ]
];

NormalForm[expr_] := expr /. gc_GC => NormalForm[gc];
S[gc_GC] := NormalForm[Total[Subsets[gc]]];
S[expr_] := Expand[expr /. gc_GC => S[gc]];
Sinv[gc_GC] := Module[
  {s = (-1)^Length[gc]},
  S[gc] /. g_GC => (-1)^Length[g] * s * g
];

Sinv[expr_] := Expand[expr /. gc_GC => Sinv[gc]];
PD[gc_GC] := Module[
  {g},
  g = FixedPoint[
    NormalForm[DeleteCases[#, Ar[a_, b_, _] /; Abs[a - b] == 1]] &,
    gc
  ];
  If[g === GC[], PD[Loop[1]],
    PD@g /. Ar[a_, b_, s_] => If[s > 0,
      X[b - 1, a, b, a - 1],
      X[b - 1, a - 1, b, a]
    ]
  ]
];

```

```

] /. 0 → 2 Length[g]
]
];
Periphery[Ar[i_, j_, s_]] := Module[
  {f = Min[i, j], p = {-i, -j*s, i, j*s}},
  While[First[p] ≠ -f, p = RotateLeft[p]];
  p
];
CC[g_GC] := Module[
  {n, gc, BlueLine, p, OtherEnds, a, i, j, i1, i2, j1, j2, a1, a2, b1, b2, t1, t2, ti, tj, s},
  n = Length[gc = NormalForm[g]];
  BlueLine = {-2 n - 1, 0};
  Scan[
    (
      p = Periphery[#];
      BlueLine = Flatten[BlueLine /. -First[p] - 1 → Rest[p]]
    ) &,
    gc
  ];
  OtherEnds = Table[0, {2 n + 2}];
  Do[
    If[(a = BlueLine[[i])] > 0,
      j = Position[BlueLine, -a - 1][[1, 1]];
      OtherEnds[[OtherEnds[[i]] = j]] = i
    ],
    {i, 2 n + 2}
  ];
  Do[
    If[(i2 = OtherEnds[[i1]]) > i1,
      {a1, a2} = BlueLine[{{i1, i2}}];
      Do[
        If[(j2 = OtherEnds[[j1]]) > j1,
          {b1, b2} = BlueLine[{{j1, j2}}];
          If[j1 < i2 < j2,
            ti = If[a1 > 0, a1 + 0.5 + 0.01 * j1, a2 + 0.5 - 0.01 * j1];
            tj = If[b1 > 0, b1 + 0.01 i2, b2 + 1 - 0.01 i2];
            {t1, t2} = Sort[{ti, tj}];
            s = Sign[ti - tj] Sign[a1] Sign[b1];
            AppendTo[gc, Ar[t1, t2, s]]
          ]
        ],
        {j1, i1 + 1, 2 n + 2}
      ],
    ],
  ];
];

```

```

      {i1, 2 n + 1}
    ];
    NormalForm[gc]
  ];
  omega[nu_][gc_GC] := Expand[Sinv[gc] /. g_GC -> nu[PD[CC[g]]]];

  t = omega[Vassiliev[2]][gc]

0

CC[gc]

GC[Ar[6, 1, 1], Ar[2, 7, 1], Ar[3, 10, -1],
  Ar[4, 11, 1], Ar[5, 14, -1], Ar[8, 13, 1], Ar[9, 12, -1]]

S[gc]

GC[] + GC[Ar[1, 2, -1]] + 2 GC[Ar[1, 2, 1]] + GC[Ar[2, 1, 1]] +
  GC[Ar[1, 2, 1], Ar[3, 4, 1]] + GC[Ar[1, 3, -1], Ar[2, 4, 1]] + GC[Ar[1, 3, 1], Ar[2, 4, -1]] +
  GC[Ar[2, 1, 1], Ar[3, 4, 1]] + GC[Ar[3, 1, 1], Ar[2, 4, -1]] + GC[Ar[3, 1, 1], Ar[2, 4, 1]] +
  GC[Ar[1, 3, 1], Ar[2, 5, -1], Ar[4, 6, 1]] + GC[Ar[3, 1, 1], Ar[2, 4, 1], Ar[5, 6, 1]] +
  GC[Ar[3, 1, 1], Ar[2, 5, -1], Ar[4, 6, 1]] + GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[3, 6, -1]] +
  GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[3, 7, -1], Ar[6, 8, 1]]

Sinv[gc]

GC[] - GC[Ar[1, 2, -1]] - 2 GC[Ar[1, 2, 1]] - GC[Ar[2, 1, 1]] +
  GC[Ar[1, 2, 1], Ar[3, 4, 1]] + GC[Ar[1, 3, -1], Ar[2, 4, 1]] + GC[Ar[1, 3, 1], Ar[2, 4, -1]] +
  GC[Ar[2, 1, 1], Ar[3, 4, 1]] + GC[Ar[3, 1, 1], Ar[2, 4, -1]] + GC[Ar[3, 1, 1], Ar[2, 4, 1]] -
  GC[Ar[1, 3, 1], Ar[2, 5, -1], Ar[4, 6, 1]] - GC[Ar[3, 1, 1], Ar[2, 4, 1], Ar[5, 6, 1]] -
  GC[Ar[3, 1, 1], Ar[2, 5, -1], Ar[4, 6, 1]] - GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[3, 6, -1]] +
  GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[3, 7, -1], Ar[6, 8, 1]]

{gc, Sinv[S[gc]], S[Sinv[gc]]}

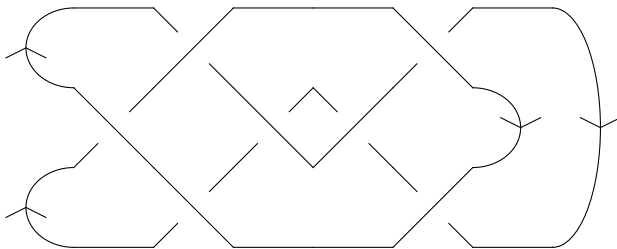
{GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[6, 8, 1], Ar[3, 7, -1]],
  GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[3, 7, -1], Ar[6, 8, 1]],
  GC[Ar[4, 1, 1], Ar[2, 5, 1], Ar[3, 7, -1], Ar[6, 8, 1]]}

DrawMorseLink [PD[CC[gc]]]

```

KnotTheory::credits: MorseLink was added to KnotTheory` by Siddarth Sankaran at the University of Toronto in the summer of 2005.

KnotTheory::credits: DrawMorseLink was written by Siddarth Sankaran at the University of Toronto in the summer of 2005.



```
Sinv[gc] /. gc_GC => PD[gc]
```

```
-PD[Loop[1]] + PD[X[2, 1, 3, 4], X[3, 1, 4, 2]] + PD[X[2, 4, 3, 1], X[3, 2, 4, 1]] +
PD[X[4, 3, 1, 2], X[3, 1, 4, 2]] - PD[X[2, 1, 3, 6], X[4, 1, 5, 2], X[5, 4, 6, 3]] -
PD[X[6, 3, 1, 2], X[4, 1, 5, 2], X[5, 4, 6, 3]] - PD[X[6, 4, 1, 3], X[4, 2, 5, 1], X[5, 2, 6, 3]] +
PD[X[8, 4, 1, 3], X[4, 2, 5, 1], X[6, 2, 7, 3], X[7, 6, 8, 5]]
```

```
Thread[
```

```
Diagrams[2 Ar] -> (omega[Vassiliev[2]][#] & /@ Diagrams[2 Ar])
```

```
]
```

```
{GC[Ar[1, 2, 1], Ar[3, 4, 1]] -> 0, GC[Ar[1, 2, 1], Ar[4, 3, 1]] -> 0,
GC[Ar[1, 2, 1], Ar[3, 4, -1]] -> 0, GC[Ar[1, 2, 1], Ar[4, 3, -1]] -> 0,
GC[Ar[2, 1, 1], Ar[3, 4, 1]] -> 0, GC[Ar[2, 1, 1], Ar[4, 3, 1]] -> 0,
GC[Ar[2, 1, 1], Ar[3, 4, -1]] -> 0, GC[Ar[2, 1, 1], Ar[4, 3, -1]] -> 0,
GC[Ar[1, 2, -1], Ar[3, 4, 1]] -> 0, GC[Ar[1, 2, -1], Ar[4, 3, 1]] -> 0,
GC[Ar[1, 2, -1], Ar[3, 4, -1]] -> 0, GC[Ar[1, 2, -1], Ar[4, 3, -1]] -> 0,
GC[Ar[2, 1, -1], Ar[3, 4, 1]] -> 0, GC[Ar[2, 1, -1], Ar[4, 3, 1]] -> 0,
GC[Ar[2, 1, -1], Ar[3, 4, -1]] -> 0, GC[Ar[2, 1, -1], Ar[4, 3, -1]] -> 0,
GC[Ar[1, 3, 1], Ar[2, 4, 1]] -> 0, GC[Ar[1, 3, 1], Ar[4, 2, 1]] -> 1,
GC[Ar[1, 3, 1], Ar[2, 4, -1]] -> 0, GC[Ar[1, 3, 1], Ar[4, 2, -1]] -> -1,
GC[Ar[3, 1, 1], Ar[2, 4, 1]] -> 0, GC[Ar[3, 1, 1], Ar[4, 2, 1]] -> 0,
GC[Ar[3, 1, 1], Ar[2, 4, -1]] -> 0, GC[Ar[3, 1, 1], Ar[4, 2, -1]] -> 0,
GC[Ar[1, 3, -1], Ar[2, 4, 1]] -> 0, GC[Ar[1, 3, -1], Ar[4, 2, 1]] -> -1,
GC[Ar[1, 3, -1], Ar[2, 4, -1]] -> 0, GC[Ar[1, 3, -1], Ar[4, 2, -1]] -> 1,
GC[Ar[3, 1, -1], Ar[2, 4, 1]] -> 0, GC[Ar[3, 1, -1], Ar[4, 2, 1]] -> 0,
GC[Ar[3, 1, -1], Ar[2, 4, -1]] -> 0, GC[Ar[3, 1, -1], Ar[4, 2, -1]] -> 0,
GC[Ar[1, 4, 1], Ar[2, 3, 1]] -> 0, GC[Ar[1, 4, 1], Ar[3, 2, 1]] -> 0,
GC[Ar[1, 4, 1], Ar[2, 3, -1]] -> 0, GC[Ar[1, 4, 1], Ar[3, 2, -1]] -> 0,
GC[Ar[4, 1, 1], Ar[2, 3, 1]] -> 0, GC[Ar[4, 1, 1], Ar[3, 2, 1]] -> 0,
GC[Ar[4, 1, 1], Ar[2, 3, -1]] -> 0, GC[Ar[4, 1, 1], Ar[3, 2, -1]] -> 0,
GC[Ar[1, 4, -1], Ar[2, 3, 1]] -> 0, GC[Ar[1, 4, -1], Ar[3, 2, 1]] -> 0,
GC[Ar[1, 4, -1], Ar[2, 3, -1]] -> 0, GC[Ar[1, 4, -1], Ar[3, 2, -1]] -> 0,
GC[Ar[4, 1, -1], Ar[2, 3, 1]] -> 0, GC[Ar[4, 1, -1], Ar[3, 2, 1]] -> 0,
GC[Ar[4, 1, -1], Ar[2, 3, -1]] -> 0, GC[Ar[4, 1, -1], Ar[3, 2, -1]] -> 0}
```

```
t = Thread[
```

```
Diagrams[3 Ar] -> (omega[Vassiliev[2]][#] & /@ Diagrams[3 Ar])
```

```
]
```

A very large output was generated. Here is a sample of it:

```
{GC[Ar[1, 2, 1], Ar[3, 4, 1], Ar[5, 6, 1]] -> 0, GC[Ar[1, 2, 1], Ar[3, 4, 1], Ar[6, 5, 1]] -> 0,
<<956>>, GC[Ar[6, 1, -1], Ar[5, 2, -1], Ar[3, 4, -1]] -> 0,
GC[Ar[6, 1, -1], Ar[5, 2, -1], Ar[4, 3, -1]] -> 0}
```

[Show Less](#)
[Show More](#)
[Show Full Output](#)
[Set Size Limit...](#)

```
t1 = Select[t, Last[#] ≠ 0 &]
```

```
{GC[Ar[1, 3, 1], Ar[2, 5, 1], Ar[6, 4, 1]] → -1, GC[Ar[1, 3, 1], Ar[2, 5, 1], Ar[6, 4, -1]] → 1,
GC[Ar[1, 3, 1], Ar[5, 2, 1], Ar[6, 4, 1]] → 1, GC[Ar[1, 3, 1], Ar[5, 2, 1], Ar[6, 4, -1]] → -1,
GC[Ar[1, 3, 1], Ar[2, 5, -1], Ar[6, 4, 1]] → 1, GC[Ar[1, 3, 1], Ar[2, 5, -1], Ar[6, 4, -1]] → -1,
GC[Ar[1, 3, 1], Ar[5, 2, -1], Ar[6, 4, 1]] → -1, GC[Ar[1, 3, 1], Ar[5, 2, -1], Ar[6, 4, -1]] → 1,
GC[Ar[3, 1, 1], Ar[2, 5, 1], Ar[6, 4, 1]] → -1, GC[Ar[3, 1, 1], Ar[2, 5, 1], Ar[6, 4, -1]] → 1,
GC[Ar[3, 1, 1], Ar[5, 2, 1], Ar[6, 4, 1]] → 1, GC[Ar[3, 1, 1], Ar[5, 2, 1], Ar[6, 4, -1]] → -1,
GC[Ar[3, 1, 1], Ar[2, 5, -1], Ar[6, 4, 1]] → 1, GC[Ar[3, 1, 1], Ar[2, 5, -1], Ar[6, 4, -1]] → -1,
GC[Ar[3, 1, 1], Ar[5, 2, -1], Ar[6, 4, 1]] → -1, GC[Ar[3, 1, 1], Ar[5, 2, -1], Ar[6, 4, -1]] → 1,
GC[Ar[1, 3, -1], Ar[2, 5, 1], Ar[6, 4, 1]] → -1, GC[Ar[1, 3, -1], Ar[2, 5, 1], Ar[6, 4, -1]] → 1,
GC[Ar[1, 3, -1], Ar[5, 2, 1], Ar[6, 4, 1]] → 1, GC[Ar[1, 3, -1], Ar[5, 2, 1], Ar[6, 4, -1]] → -1,
GC[Ar[1, 3, -1], Ar[2, 5, -1], Ar[6, 4, 1]] → 1,
GC[Ar[1, 3, -1], Ar[5, 2, -1], Ar[6, 4, 1]] → -1,
GC[Ar[1, 3, -1], Ar[5, 2, -1], Ar[6, 4, -1]] → 1,
GC[Ar[3, 1, -1], Ar[2, 5, 1], Ar[6, 4, 1]] → -1, GC[Ar[3, 1, -1], Ar[2, 5, 1], Ar[6, 4, -1]] → 1,
GC[Ar[3, 1, -1], Ar[5, 2, 1], Ar[6, 4, 1]] → 1, GC[Ar[3, 1, -1], Ar[5, 2, 1], Ar[6, 4, -1]] → -1,
GC[Ar[3, 1, -1], Ar[2, 5, -1], Ar[6, 4, 1]] → 1,
GC[Ar[3, 1, -1], Ar[2, 5, -1], Ar[6, 4, -1]] → -1,
GC[Ar[3, 1, -1], Ar[5, 2, -1], Ar[6, 4, 1]] → -1,
GC[Ar[3, 1, -1], Ar[5, 2, -1], Ar[6, 4, -1]] → 1}
```

```
t2 = Sinv[GC[Ar[1, 3, 1], Ar[2, 5, 1], Ar[6, 4, 1]]]
```

```
-GC[] + 2 GC[Ar[1, 2, 1]] + GC[Ar[2, 1, 1]] -
GC[Ar[1, 2, 1], Ar[4, 3, 1]] - GC[Ar[1, 3, 1], Ar[2, 4, 1]] -
GC[Ar[1, 3, 1], Ar[4, 2, 1]] + GC[Ar[1, 3, 1], Ar[2, 5, 1], Ar[6, 4, 1]]
```

```
t3 = t2 /. gc_GC => CC[gc]
```

```
-GC[] + 2 GC[Ar[1, 2, 1]] + GC[Ar[2, 1, 1]] -
GC[Ar[1, 2, 1], Ar[4, 3, 1]] - GC[Ar[1, 4, 1], Ar[5, 2, 1], Ar[3, 6, 1]] -
GC[Ar[1, 4, 1], Ar[2, 7, 1], Ar[3, 6, -1], Ar[5, 8, 1]] + GC[Ar[1, 4, 1],
Ar[2, 9, 1], Ar[3, 8, -1], Ar[12, 5, 1], Ar[6, 11, 1], Ar[7, 14, 1], Ar[10, 13, -1]]
```

```
t4 = t3 /. gc_GC => PD[gc]
```

```
PD[Loop[1]] - PD[X[3, 1, 4, 6], X[1, 5, 2, 4], X[5, 3, 6, 2]] -
PD[X[3, 1, 4, 8], X[6, 2, 7, 1], X[5, 2, 6, 3], X[7, 5, 8, 4]] + PD[X[3, 1, 4, 14], X[8, 2, 9, 1],
X[7, 2, 8, 3], X[4, 12, 5, 11], X[10, 6, 11, 5], X[13, 7, 14, 6], X[12, 9, 13, 10]]
```

```
t4 /. pd_PD => DrawMorseLink[pd]
```

