Pensieve Header: The Alexander blobs program, with conventions following the Chicago ax+b handout of http://www.math.toronto.edu/~drorbn/Talks/Chicago-1009/

For the ArrowRules, see testing at “AlexanderBlobs-U(I2D) Comparison.nb”.

```
ArrowRules = {
   Diag[hs_, lft___, ar[i_, j_], ar[k_, l_], rgt___] /;
     ! OrderedQ[{ar[i, j], ar[k, l]}] :→ Plus[
     Diag[hs, lft, ar[k, l], ar[i, j], rgt],
     Which[
      i == k, 0,
      j == l, Diag[h[i] hs, lft, ar[k, l], rgt] - Diag[h[k] hs, lft, ar[i, j], rgt],
      j == k && i == l, (
       -Diag[h[i], lft, ar[j, i], rgt] + Diag[h[j], lft, ar[i, j], rgt] -
        Diag[h[i] up[j], lft, rgt] + Diag[h[j] up[i], lft, rgt]
      ),
      j == k, -Diag[h[i] hs, lft, ar[k, l], rgt] + Diag[h[k] hs, lft, ar[i, l], rgt],
      i == l, -Diag[h[i] hs, lft, ar[k, j], rgt] + Diag[h[k] hs, lft, ar[i, j], rgt],
      True, 0
     ]
    ]
  };
If[Head[$DegreeStack] =!= List, $DegreeStack = {Infinity}];
$ModDegree = First[$DegreeStack];
SetAttributes[ModDegree, HoldRest];
ModDegree[m_, expr_] := Module[{res},
   PrependTo[$DegreeStack, $ModDegree = m];
   res = expr;
   $DegreeStack = Rest[$DegreeStack];
   $ModDegree = First[$DegreeStack];
   res
];
Deg[Diag[hs_, ars___]] := Length[{ars}] + Exponent[hs /. h[_] | up[_] → h, h];

Deg[Diag[h[1] up[2]^2, ar[2, 3]]]

4

Unprotect[NonCommutativeMultiply];
0 ** _ = 0;
_ ** 0 = 0;
(c_?(FreeQ[#, Diag] &) * a_) ** b_ := Expand[c * (a ** b)];
a_ ** (c_?(FreeQ[#, Diag] &) * b_) := Expand[c * (a ** b)];
a_Plus ** b_ := (# ** b) & /@ a;
a_ ** b_Plus := (a ** #) & /@ b;
d1_Diag ** d2_Diag /; Deg[d1] + Deg[d2] ≥ $ModDegree := 0;
Diag[hs1_, ars1___] ** Diag[hs2_, ars2___] :=
  Diag[hs1 * hs2, ars1, ars2] //. ArrowRules;
b[x_, y_] := x ** y - y ** x;

r[i_, j_] := Diag[1, ar[i, j]];
b[r[1, 2], r[1, 3]] + b[r[1, 2], r[2, 3]]

-Diag[h[1], ar[2, 3]] + Diag[h[2], ar[1, 3]]
```

```
b[r[1, 2], r[1, 3]] + b[r[1, 2], r[2, 3]] + b[r[1, 3], r[2, 3]]
```

```
0
```

```
DPower[expr_, p_Integer] /; p > 0 := NonCommutativeMultiply @@ Table[expr, {p}];
DExp[expr_] := Module[
   {total, term, k},
   k = 0;
   total = term = Diag[1];
   While[term =!= 0,
    ++k;
    total += (term = Expand[term ** expr / k])
   ];
   total
  ];
```

```
ModDegree[7, DExp[r[1, 2]]]
```

$\text{Diag}[1] + \text{Diag}[1, \text{ar}[1, 2]] + \frac{1}{2} \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 2]] + \frac{1}{6} \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2]] + \frac{1}{24} \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2]] + \frac{1}{120} \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2]] + \frac{1}{720} \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2], \text{ar}[1, 2]]$

```
ModDegree[3, DExp[r[1, 2]] ** DExp[r[1, 3]] ** DExp[r[2, 3]]]
```

$\text{Diag}[1] + \text{Diag}[1, \text{ar}[1, 2]] + \text{Diag}[1, \text{ar}[1, 3]] + \text{Diag}[1, \text{ar}[2, 3]] + \frac{1}{2} \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 2]] + \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 3]] + \text{Diag}[1, \text{ar}[1, 2], \text{ar}[2, 3]] + \frac{1}{2} \text{Diag}[1, \text{ar}[1, 3], \text{ar}[1, 3]] + \text{Diag}[1, \text{ar}[1, 3], \text{ar}[2, 3]] + \frac{1}{2} \text{Diag}[1, \text{ar}[2, 3], \text{ar}[2, 3]]$

```
t1 = ModDegree[3, DExp[r[2, 3]] ** DExp[r[1, 3]]]
```

$\text{Diag}[1] + \text{Diag}[1, \text{ar}[1, 3]] + \text{Diag}[1, \text{ar}[2, 3]] - \text{Diag}[h[1], \text{ar}[2, 3]] + \text{Diag}[h[2], \text{ar}[1, 3]] + \frac{1}{2} \text{Diag}[1, \text{ar}[1, 3], \text{ar}[1, 3]] + \text{Diag}[1, \text{ar}[1, 3], \text{ar}[2, 3]] + \frac{1}{2} \text{Diag}[1, \text{ar}[2, 3], \text{ar}[2, 3]]$

```
ModDegree[3, t1 ** DExp[r[1, 2]]]
```

$\text{Diag}[1] + \text{Diag}[1, \text{ar}[1, 2]] + \text{Diag}[1, \text{ar}[1, 3]] + \text{Diag}[1, \text{ar}[2, 3]] + \frac{1}{2} \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 2]] + \text{Diag}[1, \text{ar}[1, 2], \text{ar}[1, 3]] + \text{Diag}[1, \text{ar}[1, 2], \text{ar}[2, 3]] + \frac{1}{2} \text{Diag}[1, \text{ar}[1, 3], \text{ar}[1, 3]] + \text{Diag}[1, \text{ar}[1, 3], \text{ar}[2, 3]] + \frac{1}{2} \text{Diag}[1, \text{ar}[2, 3], \text{ar}[2, 3]]$

```
ModDegree[7, DExp[r[1, 2]] ** DExp[r[1, 3]] ** DExp[r[2, 3]] -
  DExp[r[2, 3]] ** DExp[r[1, 3]] ** DExp[r[1, 2]]]
```

```
0
```

```
Adjoint[Diag[hs_, ars___]] := Times[
   hs /. {h[_] :→ 1, up[_] :→ -1},
   (-1)^Length[{ars}],
   Reverse[Diag[ars, hs]]
  ];
Adjoint[expr_] := Expand[expr /. diag_Diag :→ Adjoint[diag]];
Cap[Diag[hs_]] := (hs /. {_h :→ 1, _up :→ 0}) * Diag[hs];
Cap[Diag[_, __]] := 0;
Cap[expr_] := Expand[expr /. diag_Diag] :→ Cap[diag];
```