

Pensieve header: The Manturov representation for virtual braids.

```

p[lft___, a_, -a_, rgt___] := p[lft, rgt];
p[lft___, -a_, a_, rgt___] := p[lft, rgt];
p[lft___, p[mid___], rgt___] := p[lft, mid, rgt]
Unprotect[NonCommutativeMultiply];
x_p ** y_p := x~Join~y;
p /: -x_p := Reverse[-# & /@ x];
p[x___] // Mor[rules___] := p@@ ({x} /. {rules});
l_List // m_Mor := m /@ l;

p[a, b, c, -c, d, e, a, -a, a, -b, b]
p[a, b, d, e, a]

p[b, a, -a, -b]
p[]

p /: -x_p := Reverse[-# & /@ x]
-p[a, b, c, -c, d, e, a, -a, a, -b, b]
p[-a, -e, -d, -b, -a]

-p[a, b, c, -c, d, e, a, -a, a, -b, b]
p[-a, -e, -d, -b, -a]

p[a, b, c, -c, d, e, a, -a, a, -b, b] ** (-p[a, b, c, -c, d, e, a, -a, a, -b, b])
p[]

p[a, b, d, e, -a] // Mor[a → p[a, b], e → p[f, -g]]
p[a, b, b, d, f, -g, -b, -a]

t1 = Mor[a → p[q, b, -q], b → p[-q, a, q]];
t2 = Mor[b → p[q, c, -q], c → p[-q, b, q]];
s1 = Mor[a → p[a, b, -a], b → a];
s1i = Mor[a → b, b → p[-b, a, b]];
s2 = Mor[b → p[b, c, -b], c → b];
s2i = Mor[b → c, c → p[-c, b, c]];

{p[a], p[b]} // s1 // s1i
{p[a], p[b]}

{p[a], p[b], p[c]} // s1 // s2 // s1
{p[a, b, c, -b, -a], p[a, b, -a], p[a]}

{p[a], p[b], p[c]} // s2 // s1 // s2
{p[a, b, c, -b, -a], p[a, b, -a], p[a]}

```

Overcrossings commute:

{p[a], p[b], p[c]} // t2 // s1 // s2 // t1 // s2i // s1i

{p[q, c, -q, -q, b, q, -c, -b, a, b, c, -q, -b, q, q, -c, -q], p[b], p[c]}

Undercrossings commute:

{p[a], p[b], p[c]} // t2 // s1i // s2i // t1 // s2 // s1

{p[a], p[q, -a, -q, a, b, -a, q, a, -q], p[-q, -a, q, a, c, -a, -q, a, q]}

**p[c] // t2 // s1 // s2 // t1 // s2i // s1i // t2 // s1i // s2i // t1 // s2 //
s1 // s1 // s2 // t1 // s2i // s1i // t2 // s1i // s2i // t1 // s2 // s1 // t2**

p[-q, -a, q, a, -q, b, q, -a, -q, -q, a, q, c, -q, -a, q, a, -q, -a, q, a, -q, -b,
q, -a, -q, a, q, q, -a, -q, a, q, -c, -q, -a, q, a, -q, -a, q, -a, -q, a, q, c,
-q, -a, q, a, -q, -q, -a, q, a, -q, b, q, -a, -q, a, q, -a, -q, a, q, -c, -q,
-a, q, q, a, -q, -b, q, -a, -q, a, q, -a, q, a, -q, b, q, -a, -q, -q, a, q, c,
-q, -a, q, a, -q, -a, q, a, -q, -b, q, -a, -q, a, q, q, -a, -q, a, q, -c, -q,
-a, q, a, -q, a, q, -a, -q, a, q, c, -q, -a, q, a, -q, -a, q, a, -q, b, q,
-a, -q, a, q, -a, -q, a, q, -c, -q, -a, q, q, a, -q, -b, q, -a, -q, -q, a, q,
c, -q, -a, q, q, a, -q, b, q, -a, -q, -q, a, q, c, -q, -a, q, a, -q, -a, q, a,
-q, -b, q, -a, -q, a, q, q, -a, -q, a, q, -c, -q, -a, q, a, -q, -a, q, -a, -q,
a, q, c, -q, -a, q, a, -q, -q, -a, q, a, -q, b, q, -a, -q, a, q, -a, -q, a, q,
-c, -q, -a, q, q, a, -q, -b, q, -a, -q, a, -q, -a, q, a, -q, b, q, -a, -q, -q,
a, q, c, -q, -a, q, a, -q, -a, q, a, -q, -b, q, -a, -q, a, q, q, -a, -q, a, q,
-c, -q, -a, q, a, -q, a, q, -a, -q, a, q, c, -q, -a, q, a, -q, -q, -a, q, a, -q,
b, q, -a, -q, a, q, -a, -q, a, q, -c, -q, -a, q, q, a, -q, -b, q, -a, -q, a, q]