

```

base = 100;
basef = 1000;
ij[alpha_Integer] := IntegerDigits[alpha, base, 2];
ijf[alpha_Integer] := IntegerDigits[alpha, basef, 2];
alpha[{i_, j_}] := base * i + j;
alpha[i_, j_] := base * i + j;
alphaf[{i_, j_}] := basef * i + j;
alphaf[i_, j_] := basef * i + j;
b[e[alpha1_], e[alpha2_]] := b[e[alpha1], e[alpha2]] = Module[
  {i, j, k, l},
  {i, j} = If[alpha1 < 1000, ij[alpha1], ijf[alpha1]];
  {k, l} = If[alpha2 < 1000, ij[alpha2], ijf[alpha2]];
  If[j == k && i == 1, (1/2) e[alpha[i, i]] +
    (1/2) e[alphaf[i, i]] - (1/2) e[alpha[j, j]] - (1/2) e[alphaf[j, j]], 0] +
    If[j == k && i != 1, e[alpha[i, l]], 0] - If[i == 1 && j != k, e[alpha[k, j]], 0]
];
Unprotect[NonCommutativeMultiply];
_**0 = 0;
0**_ = 0;
(a_Plus)**w_RW := (#**w) & /@ a;
e[alpha_]** (a_Plus) := (e[alpha]**#) & /@ a;
(c_*e[alpha_])**w_RW := Expand[c*(e[alpha]**w)];
e[alpha_]** (c_*w_RW) := Expand[c*(e[alpha]**w)];
e[alpha_]**RW[] := RW[alpha];
e[alpha_]**RW[beta_, rest___] /; alpha <= beta := RW[alpha, beta, rest];
e[alpha_]**RW[beta_, rest___] /; alpha > beta := Plus[
  e[beta]**(e[alpha]**RW[rest]),
  b[e[alpha], e[beta]]**RW[rest]
];

CanonicalForm[expr_] := Expand[expr /. {
  W[] -> RW[],
  W[alpha_, rest___] -> e[alpha]**CanonicalForm[W[rest]]
}];

UDGL2[diag_Diag] := Module[
  {PutAt, w},
  Expand[CanonicalForm[
    Distribute[diag /. ar[i_, j_] ->
      1/2 PutAt[i, j, 101, 1001] + PutAt[i, j, 102, 201] + 1/2 PutAt[i, j, 202, 2002]
    ]
  ]
];

```

```

] /. d_Diag => Times[
  Times @@ (d /. _PutAt -> 1),
  (
    w = Table[0, {2 Length[d]}];
    d /. PutAt[i_, j_, alpha1_, alpha2_] => (
      w[[i]] = alpha1; w[[j]] = alpha2
    );
    W@@w
  )
]
]]
];
UDGL2[expr_] := Expand[expr /. diag_Diag => UDGL2[diag]];

UDGL3[diag_Diag] := Module[
  {PutAt, w},
  Expand[CanonicalForm[
    Distribute[diag /. ar[i_, j_] => 1/2 PutAt[i, j, 101, 1001] +
      1/2 PutAt[i, j, 202, 2002] + 1/2 PutAt[i, j, 303, 3003] +
      PutAt[i, j, 102, 201] + PutAt[i, j, 103, 301] + PutAt[i, j, 203, 302]
    ] /. d_Diag => Times[
      Times @@ (d /. _PutAt -> 1),
      (
        w = Table[0, {2 Length[d]}];
        d /. PutAt[i_, j_, alpha1_, alpha2_] => (
          w[[i]] = alpha1; w[[j]] = alpha2
        );
        W@@w
      )
    ]
  ]
]]
];
UDGL3[expr_] := Expand[expr /. diag_Diag => UDGL3[diag]];

```

```

UDGL4[diag_Diag] := Module[
  {PutAt, w},
  Expand[CanonicalForm[
    Distribute[
      diag /. ar[i_, j_] => 1/2 PutAt[i, j, 101, 1001] + 1/2 PutAt[i, j, 202, 2002] +
        1/2 PutAt[i, j, 303, 3003] + 1/2 PutAt[i, j, 404, 4004] + PutAt[i, j, 102, 201] +
        PutAt[i, j, 103, 301] + PutAt[i, j, 104, 401] +
        PutAt[i, j, 203, 302] + PutAt[i, j, 204, 402] + PutAt[i, j, 304, 403]
    ] /. d_Diag => Times[
      Times @@ (d /. _PutAt -> 1),
      (
        w = Table[0, {2 Length[d]}];
        d /. PutAt[i_, j_, alpha1_, alpha2_] => (
          w[[i]] = alpha1; w[[j]] = alpha2
        );
        W@@w
      )
    ]
  ]];
UDGL4[expr_] := Expand[expr /. diag_Diag => UDGL4[diag]];

UDGL5[diag_Diag] := Module[
  {PutAt, w},
  Expand[CanonicalForm[
    Distribute[
      diag /. ar[i_, j_] => 1/2 PutAt[i, j, 101, 1001] + 1/2 PutAt[i, j, 202, 2002] + 1/2
        PutAt[i, j, 303, 3003] + 1/2 PutAt[i, j, 404, 4004] + 1/2 PutAt[i, j, 505, 5005] +
        PutAt[i, j, 102, 201] + PutAt[i, j, 103, 301] + PutAt[i, j, 104, 401] +
        PutAt[i, j, 105, 501] + PutAt[i, j, 203, 302] + PutAt[i, j, 204, 402] + PutAt[i, j,
        205, 502] + PutAt[i, j, 304, 403] + PutAt[i, j, 305, 503] + PutAt[i, j, 405, 504]
    ] /. d_Diag => Times[
      Times @@ (d /. _PutAt -> 1),
      (
        w = Table[0, {2 Length[d]}];
        d /. PutAt[i_, j_, alpha1_, alpha2_] => (
          w[[i]] = alpha1; w[[j]] = alpha2
        );
        W@@w
      )
    ]
  ]];
UDGL5[expr_] := Expand[expr /. diag_Diag => UDGL5[diag]];

```