

```

DeclareAlgebra[U_Symbol, opts_Rule] := Module[
{gp, sr, g, cp, M, pow, k = 0,
  gs = Generators /. {opts},
  cs = Centrals /. {opts} /. Centrals → {}
},
U[Generators] = gs;
(#U = U@#) & /@ gs;
gp = Alternatives @@ gs; gp = gp | gp_ (* gens *)
sr = Flatten@Table[{g → ++k, gi → {i, k}}, {g, gs}]; (* sorting → *)
cp = Alternatives @@ cs; (* cents *)
SetAttributes[M, HoldRest]; M[0, _] = 0; M[a_, x_] := a x;
CE[ε] := Collect[ε, U, Expand] /. ex → eExpand[x] /. $trim;
Ui[ε] := ε /. {t : cp ⇒ ti, uU ⇒ (#i &) /@ u};
Ui[NCM[]] = pow[ε, 0] = U@{} = 1U = U[];
B[U@(x-)i, U@(y-)i] := Ui@B[U@x, U@y];
B[U@(x-)i, U@(y-)j] /; i != j := 0;
B[U@y-, U@x-] := CE[-B[U@x, U@y]];
x- ⓧ (c-. 1U) := CE[c x]; (c-. 1U) ⓧ x- := CE[c x];
(a-. U[xx___, x-]) ⓧ (b-. U[y-, yy___]) := If[OrderedQ[{x, y} /. sr],
  CE@M[a b /. $trim, U[xx, x, y, yy]],
  U@xx ⓧ CE@M[a b /. $trim, U@y ⓧ U@x + B[U@x, U@y, $E]] ⓧ U@yy];
U@{c-. * (L : gp)n_, r___} /; FreeQ[c, gp] := CE[c U@Table[L, {n}] ⓧ U@{r}];
U@{c-. * L : gp, r___} := CE[c U[L] ⓧ U@{r}];
U@{c-, r___} /; FreeQ[c, gp] := CE[c U@{r}];
U@{L_Plus, r___} := CE[U@{#, r} & /@ L];
U@{L-, r___} := U@{Expand[L], r};
U[ε_NonCommutativeMultiply] := U /@ ε;
OU[specs___, poly_] := Module[{sp, null, vs, us},
  sp = Replace[{specs}, L_List ⇒ Lnull, {1}];
  vs = Join @@ (First /@ sp);
  us = Join @@ (sp /. Ls ⇒ (L /. xi ⇒ xs));
  CE[Total[
    CoefficientRules[poly, vs] /. (p- → c-) ⇒ c U@(usp)
  ] /. xnull ⇒ x];
OU[specs___, E[L-, Q-, P-]] := OU[specs, SS@Normal[P eL+Q]];
pow[ε, n_] := pow[ε, n - 1] ⓧ ε;
SU[ε, ss___Rule] := CE@Total[
  CoefficientRules[ε, First /@ {ss}] /. (p- → c-) ⇒ c NCM @@ MapThread[pow, {Last /@ {ss}, p}]]];
σrs[[c-. * uU] := (c /. (t : cp)j ⇒ tj /. {rs}) U[List @@ (u /. v-j ⇒ vj /. {rs})];
mj → k[[c-. * uU] :=
  CE[(c /. (t : cp)j → tk) DeleteCases[u, _j|k] ⓧ U @@ Cases[u, w-j ⇒ wk] ⓧ U @@ Cases[u, _k]];
U /: c-. * uU * vU := CE[c u ⓧ v];
Si[[c-. * uU] :=
  CE[(c /. Si[U, Centrals]) DeleteCases[u, _i] ⓧ Ui[NCM @@ Reverse@Cases[u, x-i ⇒ S@U@x]]];
Δi → j, k[[c-. * uU] :=
  CE[(c /. Δi → j, k[U, Centrals]) DeleteCases[u, _i] ⓧ
    (NCM @@ Cases[u, x-i ⇒ σ1 → j, 2 → k@Δ@U@x] /. NCM[] → U[])];

```