```
DeclareAlgebra[U_Symbol, opts__Rule] :=
 Module[{gp, sr, g, cp, M, CE, k = 0,
   gs = Generators /. {opts},
   cs = Centrals /. {opts} /. Centrals → {} },
  (#U = U@#) & /@ gs;
  gp = Alternatives @@ gs; gp = gp | gp_; (* gens *)
  sr = Flatten@Table[{g → ++k, g_i_ → {i, k}}, {g, gs}];
  (* sorting → *)
  cp = Alternatives @@ cs; (* cents *)
  SetAttributes[M, HoldRest]; M[0, _] = 0;
  M[a_, x_] := a x;
  CE[ℰ_] := Collect[ℰ, _U, Expand] /. $trim;
  U_i_[ℰ_] := ℰ /. {t : cp :↦ t_i, u_U :↦ (#_i &) /@ u};
  U_i_[NCM[]] = U@{} = 1_U = U[];
  B[U@(x_)_i_, U@(y_)_i_] := U_i@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_. 1_U) := CE[c x]; (c_. 1_U) ** 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[ℰ_NonCommutativeMultiply] := U /@ ℰ;
  𝕆_U[specs___, poly_] := Module[{sp, null, vs, us},
    sp = Replace[{specs}, l_List :↦ l_null, {1}];
    vs = Join @@ (First /@ sp);
    us = Join @@ (sp /. l_s_ :↦ (l /. x_i_ :↦ x_s));
    CE[Total[
       CoefficientRules[poly, vs] /. (p_ → c_) :↦ c U@(us^p)
      ]] /. x_null :↦ x];
  𝕆_U[specs___, 𝔼[L_, Q_, P_]] :=
   𝕆_U[specs, SS@Normal[P ℮^(L+Q)]];
  σrs___[c_. * u_U] :=
   (c /. (t : cp)_j_ :↦ t_j/.{rs}) U[List @@ (u /. v_j_ :↦ v_j/.{rs})];
  m_j_→k_[c_. * u_U] :=
   CE[((c /. (t : cp)_j → t_k) DeleteCases[u, _j|k]) **
     U @@ Cases[u, w_j :↦ w_k] ** U @@ Cases[u, _k]];
  U /: c_. * u_U * v_U := CE[c u ** v];
  S_i_[c_. * u_U] :=
   CE[((c /. S_i[U, Centrals]) DeleteCases[u, _i]) **
     U_i[NCM @@ Reverse@Cases[u, x_i :↦ S@U@x]]];
  Δ_i_→j_,k_[c_. * u_U] :=
   CE[((c /. Δ_i→j,k[U, Centrals]) DeleteCases[u, _i]) **
     (NCM @@ Cases[u, x_i :↦ σ_1→j,2→k@Δ@U@x] /.
       NCM[] → U[]); ]
```