Pensieve Header: An implementation of the Free Associative Algebra (FAA) universe of operations.

We want to implement: $\sigma_{i\to j}$, $m_{i,j\to k}$, $\partial[v]_{i\to j,k}$, $S_{i\to j}$, $\Delta_{i\to j,k}$, $\mathrm{tr}_{i\to j}$, maybe ϵ and η, $\otimes$.

Data format: $\mathrm{AW}_i[x, y, x]\,\mathrm{AW}_j[y, x, y]\,\mathrm{CW}_k[x, x, x, y]$ ... (denotes an element of overall degree 10.

```
<< C:\drorbn\AcademicPensieve\People\Kuno\Profile.m
```

```
(* PP_=Identity *)
```

Cyclic Words and the Trace

In[•]:=
```
RotateToMinimal[aw_] := Module[
   {best = aw, rotated = RotateLeft[aw]},
   While[rotated =!= aw,
    best = First[Sort[{best, rotated}]];
    rotated = RotateLeft[rotated]
   ];
   best
  ];
```

In[•]:=
```
CF[CW_i_[iw___]] := RotateToMinimal@CW_i[iw];
tr_i_→j_[ℰ_] := ℰ /. AW_i[iw___] :> CF@CW_j[iw]
```

Bases

In[•]:=
```
Basis_d_,ps_[AW_i_] := AW_i @@@ Tuples[ps, d]
Basis_d_,ps_[AW_i_ more_] :=
  Flatten@Table[Outer[Times, Basis_d1,ps[AW_i], Basis_d-d1,ps[more]], {d1, 0, d}];
Basis_d_,ps_[CW_i_] := Union[tr_i→i /@ Basis_d,ps[AW_i]];
Basis_d_,ps_[CW_i_ more_] :=
 Flatten@Table[Outer[Times, Basis_d1,ps[CW_i], Basis_d-d1,ps[more]], {d1, 0, d}]
```

Factor Renaming

```
σ_i_→j_[ℰ_] := PP_FAAσ[ℰ /. {AW_i[iw___] :> AW_j[iw], CW_i[iw___] :> CW_j[iw]}]
```

Multiplication

```
m_i_,j_→k_[ℰ_] := PP_FAAm[ℰ /. AW_i[iw___] AW_j[jw___] :> AW_k[iw, jw]]
```

Word "Cutting"

```
𝔻[v_]i_→j_,k_[ℰ_] := PPFA𝔻@Expand[ℰ /. {
      AWi[] → 0,
      AWi[iw__] :↦ Sum[
        If[{iw}⟦α⟧ === v, Take[AWj[iw], α - 1] × Drop[AWk[iw], α], 0], {α, Length@{iw}}]
    }];
𝔻[v_]i_→j_[ℰ_] := PPFA𝔻@Expand[ℰ /. {
     CWi[] → 0,
     CWi[iw__] :↦
      Sum[If[{iw}⟦α⟧ === v, Most@RotateLeft[AWj[iw], α], 0], {α, Length@{iw}}]
   }]
```

The Co-Product

```
Δi_→j_,k_[ℰ_] := PPFAΔ@Expand@Module[{αs, all},
    ℰ /. {
      AWi[iw___] :↦ (
        all = Range@Length@{iw};
        Sum[
         AWj[iw]⟦αs⟧ AWk[iw]⟦Complement[all, αs]⟧,
         {αs, Subsets@all}
        ]),
      CWi[iw___] :↦ (
        all = Range@Length@{iw};
        Sum[
         CF[CWj[iw]⟦αs⟧] × CF[CWk[iw]⟦Complement[all, αs]⟧],
         {αs, Subsets@all}
        ])
     }
   ]
```

The Antipode

```
Si_→j_[ℰ_] := PPFAS@Expand[
   ℰ /. {
     AWi[iw___] :↦ (-1)^Length@{iw} Reverse[AWj[iw]],
     CWi[iw___] :↦ (-1)^Length@{iw} CF@Reverse[CWj[iw]]
    }
  ]
```

The Unit

In[•]:=
```
η_i_[ℰ_] := Expand[ℰ AW_i[]]
```

Substitutions

Currently limited to sending variables to sums of others, with no coefficients.

```
FA[rs___Rule][ℰ_] :=
 PP_FAFA@Expand[ℰ /. AW_i_[iw___] :> Distribute[AW_i[iw] /. {rs} /. AW_i[___, 0, ___] → 0]]
```

Exterior Multiplication

The built in Series method was very slow:

```
(*EM_d_[ℰ1_,ℰ2_]:=PP_FAEM@Expand[Normal[Times[
        PP_Cropping[PP_CompDeg@
            (ℰ1/.{AW_i_[iw__]:>ħ^Length@{iw} AW_i[iw],CW_i_[iw__]:>ħ^Length@{iw} CW_i[iw]})+O[ħ]^(d+1)],
        PP_Cropping[PP_CompDeg@
            (ℰ2/.{AW_i_[iw__]:>ħ^Length@{iw} AW_i[iw],CW_i_[iw__]:>ħ^Length@{iw} CW_i[iw]})+O[ħ]^(d+1)]
      ]+O[ħ]^(d+1)]/.ħ→1]*)
```

```
EM_d_[ℰ1_, ℰ2_] := PP_FAEM@Module[{E1, E2, j, j1},
   E1 = CoefficientList[
     ℰ1 /. {AW_i_[iw__] :> ħ^Length@{iw} AW_i[iw], CW_i_[iw__] :> ħ^Length@{iw} CW_i[iw]}, ħ];
   E2 = CoefficientList[
     ℰ2 /. {AW_i_[iw__] :> ħ^Length@{iw} AW_i[iw], CW_i_[iw__] :> ħ^Length@{iw} CW_i[iw]}, ħ];
   Sum[
    Expand[E1[[j1 + 1]] E2[[j - j1 + 1]]],
    {j, 0, d}, {j1, 0, j}]
  ]
```

In[•]:=
```
? CoefficientList
```

Out[•]=

Symbol

CoefficientList[*poly*, *var*] gives a list of coefficients of powers of *var* in *poly*, starting with power 0.

CoefficientList[*poly*, {var_1, var_2, …}] gives an array of coefficients of the var_i.

CoefficientList[*poly*, {var_1, var_2, …}, {dim_1, dim_2, …}] gives an

array of dimensions {dim_1, dim_2, …}, truncating or padding with zeros as needed.

In[•]:=
```
CoefficientList[2 ħ + 3 ħ^2, ħ]
```

Out[•]=
```
{0, 2, 3}
```