Conventions on August 18

Only one algebra (not necessarily Lie!) at a time, and it comes with a list of (Latin) generators and their Greek duals, and a default PBW order. If some Verma module is needed, it is added once and by means of extra generators.
Generators are of the form $x_{\alpha,\beta}[i]$, $a_\alpha[i]$, etc., where i is the tensor factor.
In the commutative context we have $\mathbb{E}_{i,j\to k}[L_0, L_1, \ldots]$, where L_p is the coefficient of ϵ^p, and (maybe) $\mathbb{Q}_{,j\to k}[F]$.
In the non-commutative context, generators do not carry a "tensor factor" designations. Instead we consider expressions like $U_i[x, y, y, x]$.

Issues: Upper and lower?

Experiments

```
In[•]:= e^a e^b
Out[•]=
        e^(a+b)
```

Conventions on August 17

Generators can be any Symbol (e.g. x,y,z,a,b,t,x12...). We use Format and Interpretation to make the appearances prettier, as in x12 → x_{12}.
Latin to Greek duality is global: $x^* = \xi$, etc.
In the non-commutative context, generators do not carry a "tensor factor" designations. Instead we consider expressions like
$U1_i[x, y, y, x]$

Experiments

```
In[•]:= U1@i[x, y, y, x]
Out[•]=
        U1[i[x, y, y, x]]

In[•]:= Format[x12, StandardForm] = x12
Out[•]=
        x1,2

In[•]:= {x13@3, x12_3}
Out[•]=
        {x13[3], x1,2_3}
```

```
In[•]:= Do[
         Format[Evaluate@Symbol["x" <> ToString[i] <> ToString[j]], StandardForm] = x10 i+j,
         {i, 3}, {j, 3}
        ]
```

```
In[•]:= x234
```

```
Out[•]=
```

$$x_{2,3_4}$$

```
In[•]:= Module[{x},
         x_i_,j_ := Symbol["x" <> ToString[i] <> ToString[j]];
         Do[
          Format[x_i,j, StandardForm] = ToExpression[SubscriptBox["x", ToString@i <> ToString@j]],
          {i, 3}, {j, 3}
         ];
         Do[
          B[x_i,j, x_k,l] = x_i,l,
          {i, 3}, {j, 3}, {k, 3}, {l, 3}
         ]
        ]
```

```
In[•]:= B[x12, x23]4
```

```
Out[•]=
```

$$x_{134}$$

```
In[•]:= x1,2
```

```
Out[•]=
```

xij

```
In[•]:= ToBoxes[y1,2]
```

```
Out[•]=
```

SubscriptBox[y, RowBox[{1, ,, 2}]]

```
In[•]:= ToExpression[SubscriptBox["y", RowBox[{1, ",", "2"}]]]
```

```
Out[•]=
```

$$y_{1,2}$$