Normal Ordering and Commutator

定义

示例

表达式的输入和输出

使用 $a[2]$ 和a[2]† 表示第二个模式的湮灭、产生算符，
算符乘积应该使用非交换乘法NonCommutativeMultiply
(**)，也可以直接使用幂次

$3\, a_2^{\dagger 2}\, a_1^3 + 5\, a_1^\dagger$

In[43]:= **3 a[2]$^{\dagger 2}$ ** a[1]^3 + 5 a[1]†**

Out[43]=

3 op[−2] ** op[−2] ** op[1] ** op[1] ** op[1] + 5 op[−1]

$(a_1 + a_1^\dagger)^2$

In[44]:= **(a[1] + a[1]†)^2**

Out[44]=

op[−1] ** op[−1] + op[−1] ** op[1] + op[1] ** op[−1] + op[1] ** op[1]

非交换乘法合并 col (NonCommutativeMultiply Collect)可以将非交换乘法合并成幂次
非交换乘法展示 dis (NonCommutativeMultiply Display)可以美化输出结果

In[45]:= **% // col // dis**

Out[45]=

$a_1\, a_1^\dagger + a_1^\dagger\, a_1 + a_1{}^2 + a_1^{\dagger 2}$

In[46]:= **%% // dis**

Out[46]=

$a_1 a_1 + a_1 a_1^\dagger + a_1^\dagger a_1 + a_1^\dagger a_1^\dagger$

化简函数 sim (simplify)可以化简NonCommutativeMultiply[1]， 通常不需要使用这个函数，因为表达式已经自动化简

In[∘]:=
```
NonCommutativeMultiply[x]
% // sim
```

Out[∘]=

NonCommutativeMultiply[x]

Out[∘]=

x

自动完成非交换乘法的相关运算，例如非算符作为系数提到非交换乘法外面

In[∘]:=
```
a[2]† ** (t a[2])
```

Out[∘]=

t op[−2] ** op[2]

对易子的计算

使用对易子 com (commutator) 计算算符的对易子

$[a_2, a_3^\dagger]$

$[a_2^\dagger, a_2]$

$[a_2, a_2^\dagger]$

In[∘]:=
```
com[a[2], a[3]†]
com[a[2]†, a[2]]
com[a[2], a[2]†]
```

Out[∘]=

0

Out[∘]=

−1

Out[∘]=

1

结合非交换乘法和对易子

$[a_2, 3\, a_2^{\dagger 2}\, a_1^3 + 5\, a_1^\dagger]$

In[∘]:=
```
com[a[2], 3 a[2]†^2 ** a[1]^3 + 5 a[1]†]
```

Out[∘]=

6 op[−2] ** op[1] ** op[1] ** op[1]

In[∘]:=
```
com[a[2], 3 a[2]†^2 ** a[1]^3 + 5 a[1]†] // col // dis
```

Out[∘]=

$6\, a_2^\dagger\, a_1{}^3$

算例

计算 0 至 9 阶对易子 $\left[\left(\frac{1}{2}\left(\xi^* a_1^2 - \xi a_1^{\dagger 2}\right)\right)^n, a_1\right]$

In[66]:=
```
x = 1/2 (r E^(-I θ) a[1]^2 - r E^(I θ) a[-1]^2);(*Log(S(ξ))*)
y = a[1];
adx[y_] := com[x, y];
```

In[69]:=
```
NestList[adx, y, 9] // Simplify // dis
```

Out[69]=

$$\{a_1, e^{i\theta} r a_1^\dagger, r^2 a_1, e^{i\theta} r^3 a_1^\dagger, r^4 a_1, e^{i\theta} r^5 a_1^\dagger, r^6 a_1, e^{i\theta} r^7 a_1^\dagger, r^8 a_1, e^{i\theta} r^9 a_1^\dagger\}$$

In[70]:=
```
Clear[x, y, adx]
```

使用ord (normal ordering)进行正规排序

$$:a_2 a_2^\dagger: = 1 + a_2^\dagger a_2$$

In[∘]:=
```
sim[a[2] ** a[2]†]
% // ord // dis
```

Out[∘]=

op[2] ** op[−2]

Out[∘]=

$$1 + a_2^\dagger a_2$$

$$:e^{i\phi/2}\left(a_2^\dagger a_2 - a_1^\dagger a_1\right) a_1^\dagger:$$

In[∘]:=
```
E^(I ϕ/2) (a[2]† ** a[2] - a[1]† ** a[1]) ** a[1]† // ord // sim // srt // dis
```

Out[∘]=

$$e^{\frac{i\phi}{2}}\left(-a_1^\dagger a_1^\dagger a_1 + a_1^\dagger a_2^\dagger a_2 - a_1^\dagger\right)$$

$$:a_1\left(q a_1 + p a_1^\dagger\right)^4:$$

In[47]:=
```
H = q a[1] + p a[1]†;
a[1]^2 ** H^4 // ord;
% // col // Collect[#, {p, q}] &
% // dis
```

Out[49]=

$$\begin{aligned}&p^3 q\,(6 + 18\,op[-1]^2 ** op[1]^2 + 4\,op[-1]^3 ** op[1]^3 + 20\,op[-1] ** op[1] + 6\,(1 + op[-1] ** op[1]) + \\ &\qquad 2\,(op[-1]^2 ** op[1]^2 + op[-1] ** op[1]) + 10\,(op[-1]^2 ** op[1]^2 + 2\,op[-1] ** op[1])) + \\ &\quad p^4\,(4\,op[-1]^3 ** op[1] + op[-1]^4 ** op[1]^2 + 6\,op[-1]^2 + 2\,(op[-1]^3 ** op[1] + op[-1]^2) + \\ &\qquad 2\,(op[-1]^3 ** op[1] + 2\,op[-1]^2)) + p^2 q^2\,(4\,op[-1]^2 ** op[1]^4 + 24\,op[-1] ** op[1]^3 + 27\,op[1]^2 + \\ &\qquad 2\,(op[-1]^2 ** op[1]^4 + 6\,op[-1] ** op[1]^3 + 6\,op[1]^2)) + p\,q^3\,(4\,op[-1] ** op[1]^5 + 14\,op[1]^4) + q^4\,op[1]^6\end{aligned}$$

Out[50]=

$$\begin{aligned}&p^3 q\left(6 + 18 a_1^{\dagger 2} a_1^2 + 4 a_1^{\dagger 3} a_1^3 + 20 a_1^\dagger a_1 + 6\left(1 + a_1^\dagger a_1\right) + 2\left(a_1^{\dagger 2} a_1^2 + a_1^\dagger a_1\right) + 10\left(a_1^{\dagger 2} a_1^2 + 2 a_1^\dagger a_1\right)\right) + \\ &\quad p^2 q^2\left(4 a_1^{\dagger 2} a_1^4 + 24 a_1^\dagger a_1^3 + 27 a_1^2 + 2\left(a_1^{\dagger 2} a_1^4 + 6 a_1^\dagger a_1^3 + 6 a_1^2\right)\right) + p q^3\left(4 a_1^\dagger a_1^5 + 14 a_1^4\right) + \\ &\quad q^4 a_1^6 + p^4\left(4 a_1^{\dagger 3} a_1 + a_1^{\dagger 4} a_1^2 + 6 a_1^{\dagger 2} + 2\left(a_1^{\dagger 3} a_1 + a_1^{\dagger 2}\right) + 2\left(a_1^{\dagger 3} a_1 + 2 a_1^{\dagger 2}\right)\right)\end{aligned}$$

In[63]:=
```
NonCommutativeMultiply @@@ Table[Table[a[1]†, {k, n}], {n, 1, 6}] // sim;
% // dis
(com[a[1], #] & /@ (%%)) // col // dis
```

Out[64]=

$$\{a_1^\dagger,\ a_1^\dagger a_1^\dagger,\ a_1^\dagger a_1^\dagger a_1^\dagger,\ a_1^\dagger a_1^\dagger a_1^\dagger a_1^\dagger,\ a_1^\dagger a_1^\dagger a_1^\dagger a_1^\dagger a_1^\dagger,\ a_1^\dagger a_1^\dagger a_1^\dagger a_1^\dagger a_1^\dagger a_1^\dagger\}$$

Out[65]=

$$\{1,\ 2\,a_1^\dagger,\ 3\,a_1^{\dagger 2},\ 4\,a_1^{\dagger 3},\ 5\,a_1^{\dagger 4},\ 6\,a_1^{\dagger 5}\}$$

如果存在多个模式，在ord后使用 srt (sorting) 使得下标的顺序正确

In[74]:=
```
a[2]† ** a[2] ** a[1]
% // dis
%% // srt // dis
```

Out[74]=

```
op[-2] ** op[2] ** op[1]
```

Out[75]=

$$a_2^\dagger a_2 a_1$$

Out[76]=

$$a_2^\dagger a_1 a_2$$

级数展开

In[•]:=
```
Series[E^(a[1]†), {a[1]†, 0, 3}] // Normal;
% // dis
%% // ord // col // dis
```

Out[•]=

$$1 + \frac{1}{2}\, a_1^\dagger a_1^\dagger + \frac{1}{6}\, a_1^\dagger a_1^\dagger a_1^\dagger + a_1^\dagger$$

Out[•]=

$$1 + a_1^\dagger + \frac{a_1^{\dagger 2}}{2} + \frac{a_1^{\dagger 3}}{6}$$

```
(Series[E^(t (a[1]† + a[1])), {t, 0, 3}] // Normal) /. Times → NonCommutativeMultiply /. t → 1;
(*有时Series在转化为Normal形式时使用的不是Power而是Times，
  此时需要手动将Times替换为NonCommutativeMultiply*)
% // dis
%% // ord // col // FullSimplify // dis
```

Out[•]=

$$1 + \frac{1}{2}\, a_1 a_1 + \frac{1}{2}\, a_1 a_1^\dagger + \frac{1}{2}\, a_1^\dagger a_1 + \frac{1}{2}\, a_1^\dagger a_1^\dagger + \frac{1}{6}\, a_1 a_1 a_1 + \frac{1}{6}\, a_1 a_1 a_1^\dagger + \frac{1}{6}\, a_1 a_1^\dagger a_1 + \frac{1}{6}\, a_1 a_1^\dagger a_1^\dagger + \frac{1}{6}\, a_1^\dagger a_1 a_1 + \frac{1}{6}\, a_1^\dagger a_1 a_1^\dagger + \frac{1}{6}\, a_1^\dagger a_1^\dagger a_1 + \frac{1}{6}\, a_1^\dagger a_1^\dagger a_1^\dagger + a_1 + a_1^\dagger$$

Out[•]=

$$\frac{1}{6}\left(3\, a_1^{\dagger 2} a_1 + 3\, a_1^\dagger\, a_1^{2} + 6\, a_1^\dagger\, a_1 + 9\left(1 + a_1 + a_1^\dagger\right) + 3\, a_1^{2} + a_1^{3} + 3\, a_1^{\dagger 2} + a_1^{\dagger 3}\right)$$

计算BCH公式到任意高阶

待补充

利用FindSequenceFunction进行BCH公式相关的计算

定义 $\mathrm{Ad}_X(Y) = X.Y.X^{-1}$ 和 $\mathrm{ad}_X(Y) = [X, Y]$，有公式 $\mathrm{Ad}_{e^X}(Y) = e^{\mathrm{ad}_X}(Y)$，即

$e^X\, Y\, e^{-X} = \sum_{n=0}^{\infty} \frac{[(X)^n, Y]}{n!}$。

利用内置函数FindSequenceFunction解决上式右边的求和就可以计算左边

定义函数AdExp $(X, Y) = e^X\, Y\, e^{-X}$

```
In[77]:= AdExp[x_, y_] := Block[{adx, seq, n, max}, n = 5(*设置为预计会出现零的对易子重数*);
   max = 12(*计算max重对易子，用于寻找规律的数列项数为max+1*);
   adx[z_] := com[x, z];
   seq = NestWhileList[adx, y, # =!= 0 &, n, max];
   If[Length[seq] ≥ n + 3, Sum[1/(n - 1)! FindSequenceFunction[seq][n], {n, 1, ∞}], Total[seq]]]
```

算例

计算 $S\left(r e^{i\theta}\right) a\, S\left(r e^{i\theta}\right)^\dagger$

```
In[•]:= $Assumptions = r > 0;

In[•]:= x = 1/2 (r E^(-I θ) a[1]^2 - r E^(I θ) a[-1]^2);
AdExp[x, a[1]] // ExpToTrig // FullSimplify;
% // dis
```

Out[•]=

$\mathrm{Cosh}[r]\, a_1 + e^{i\theta}\, \mathrm{Sinh}[r]\, a_1^\dagger$

```
In[•]:= AdExp[x, a[1]†] // ExpToTrig // FullSimplify;
% // dis
```

Out[•]=

$e^{-i\theta}\, \mathrm{Sinh}[r]\, a_1 + \mathrm{Cosh}[r]\, a_1^\dagger$

计算 $D(\alpha)^\dagger a\, D(\alpha)$

```
In[•]:= x = -α a[1]† + Conjugate[α] a[1];
AdExp[x, a[1]] // ExpToTrig // FullSimplify;
% // dis
```

Out[•]=

$\alpha + a_1$