In[•]:=
```
(*Clear any existing symbol definitions*)ClearAll[e, s, n, w];

(*Define rules*)+
rules = {
   (*Combine terms in the same direction*)
   e[x_] * e[y_] :> e[x + y],
   s[x_] * s[y_] :> s[x + y],
   n[x_] * n[y_] :> n[x + y],
   w[x_] * w[y_] :> w[x + y],

   (*Power rules*)
   e[x_]^n_ :> e[x * n],
   s[x_]^n_ :> s[x * n],
   n[x_]^n_ :> n[x * n],
   w[x_]^n_ :> w[x * n],

   (*Commutative property for multiplication*)
   e[x_] * s[y_] :> s[y] * e[x],
   e[x_] * n[y_] :> n[y] * e[x],
   e[x_] * w[y_] :> w[y] * e[x],
   s[x_] * n[y_] :> n[y] * s[x],
   s[x_] * w[y_] :> w[y] * s[x],
   n[x_] * w[y_] :> w[y] * n[x]
  };

(*Example expression*)
expr = e[1] * s[1] + e[1] * e[1] * s[1];

(*Apply rules*)
simplifiedExpr = expr //. rules
```
Out[•]=
```
(e[1] + e[2]) s[1]
```

In[•]:=
```
(*Clear any existing definitions*)
ClearAll[e, s, n, w];

(*Basic rules:handle multiplication and addition*)
rules = {
    (*Combine terms in the same direction*)
    e[x_] * e[y_] :> e[x + y],
    s[x_] * s[y_] :> s[x + y],
    e[x_]^n_ :> e[x * n],
    s[x_]^n_ :> s[x * n],

    (*Commutative property*)
    e[x_] * s[y_] :> s[y] * e[x],

    (*Factor out common terms*)
    a_. * s[x_] + b_. * s[x_] :> s[x] * (a + b)};

(*Example expression*)
expr = e[1] * s[1] + e[1] * e[1] * s[1];

(*Apply the rules to simplify*)
simplifiedExpr = expr //. rules
```

Out[•]=
$$(e[1] + e[2])\, s[1]$$

In[•]:=
```
(*Example expression*)
expr = e[1] ⨯ e[1] * s[1] + e[1] * e[1] * s[2] + e[2] * s[3] + s[2] + s[2] * e[1];

(*Apply the rules to simplify*)
simplifiedExpr = expr //. rules
```

Out[•]=
$$e[2] \times s[1] + (1 + e[1] + e[2])\, s[2] + e[2] \times s[3]$$