

BNF syntax for a chemical formula without parentheses

```
<upper> ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" |  
            "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" |  
            "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
```

```
<lower> ::= "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" |  
            "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" |  
            "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
```

```
<digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

```
<symbol> ::= <upper> { <lower> }
```

```
<number> ::= <digit> +
```

```
<formula> ::= ( <symbol> { <number> } ) +
```

Reminders

<XXX>

is replaced by something that matches its definition after ::=

"xxx"

appears unchanged (without the quotes)

X Y Z ...

is replaced by something produced from X followed by something that matches Y followed by something that matches Z ...

X | Y | Z ...

is replaced by either something produced from X or something that matches Y or something that matches Z ...

{ X }

is replaced by something that matches X or nothing at all

X +

is replaced by any number (more than 0) of things that match X

X *

is replaced by any number (including 0) of things that match X

(X)

normal meaning of parentheses: does X Y * mean one X followed by any number of Ys: X Y Y Y ... or is it X Y X Y X Y ...? It is the former, for the latter write (X Y) *