Chapter 3
Expressions

All objects with which *Mathematica* works are expressions. There are two classes of them—atoms and composite expressions.

3.1 Atoms

There are three kinds of atoms—numbers, symbols, and strings.

Numbers

Integer numbers (of unlimited size).
In[1] := 1234567890
Out[1] = 1234567890
A rational number consists of the numerator and the denominator.
In[2] := 1234567890/987654321
Out[2] = $\frac{137174210}{109739369}$
A complex number consists of the real and imaginary parts.
In[3] := 1 + 2 ∗ *I*
Out[3] = 1 + 2i
Real numbers can have arbitrarily high precision.
In[4] := 1234567890.987654321
Out[4] = $1.23456789098765432 \times 10^9$

A. Grozin, *Introduction to Mathematica® for Physicists*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-00894-3_3,

Symbols

A variable can be in one of two states. Initially it is free—it means itself (a symbol).
In[5] := $\boldsymbol{x}$
Out[5] = x
Assigning a value to it, we make it bound.
In[6] := $\boldsymbol{x}$ **= 123**
Out[6] = 123
Now, when we use it (e.g., just by asking *Mathematica* to print it), its value is substituted.
In[7] := $\boldsymbol{x}$
Out[7] = 123
How to make it free again?
In[8] := Clear[*x*]
Let's check.
In[9] := $\boldsymbol{x}$
Out[9] = x

Strings

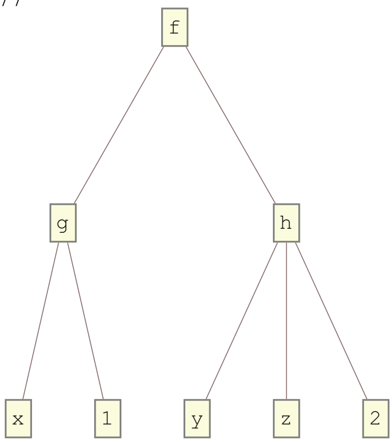
In[10] := "This is a string"
Out[10] = This is a string

3.2 Composite Expressions

A composite expression is a function of a number of arguments, each of which is an expression (i.e., an atom or a composite expression).
In[11] := $\boldsymbol{a = f[g[x,1],h[y,z,2]]}$
Out[11] = $f[g[x,1],h[y,z,2]]$
Each composite expression has a head—the function which is applied to arguments.
In[12] := Head[*a*]
Out[12] = f
The number of arguments is given by the function Length.
In[13] := Length[*a*]
Out[13] = 2
Arguments are extracted by the function Part.
In[14] := Part[*a*, 1]
Out[14] = $g[x,1]$
In[15] := Part[*a*, 2]
Out[15] = $h[y,z,2]$
And this is the first part of the second part of the expression a.

In[16] := Part[*a*, 2, 1]
Out[16] = *y*
An alternative syntax.
In[17] := *a*[[2, 1]]
Out[17] = *y*
Zeroth part of an expression is its head.
In[18] := Part[*a*, 0]
Out[18] = *f*
By the way, a head can be any expression, not just a symbol.
In[19] := *b* = *f*[*x*][*y*, 1]
Out[19] = *f*[*x*][*y*, 1]
In[20] := Head[*b*]
Out[20] = *f*[*x*]
Expressions are trees whose leaves are atoms.
In[21] := TreeForm[*a*]

Out[21]//TreeForm =



Parts of an expressions can be changed.
In[22] := *a*[[1, 2]] = 0; *a*
Out[22] = *f*[*g*[*x*, 0], *h*[*y*, *z*, 2]]
In[23] := *a*[[0]] = *j*; *a*
Out[23] = *j*[*g*[*x*, 0], *h*[*y*, *z*, 2]]
A group of arguments can be selected, not just a single argument.
In[24] := *b* = *f*[x1, x2, x3, x4, x5, x6]
Out[24] = *f*[x1, x2, x3, x4, x5, x6]
In[25] := Part[*b*, Span[2, 4]]
Out[25] = *f*[x2, x3, x4]
An alternative syntax.
In[26] := *b*[[2;; 4]]
Out[26] = *f*[x2, x3, x4]

From the beginning to 3:
In[27] := *b*[[;;3]]
Out[27] = *f*[x1, x2, x3]
From 4 to the end:
In[28] := *b*[[4;;]]
Out[28] = *f*[x4, x5, x6]
From 1 to 5 by 2:
In[29] := *b*[[1;;5;;2]]
Out[29] = *f*[x1, x3, x5]
If such a form is used in the left-hand side of an assignment, each of the selected arguments will be replaced:
In[30] := *b*[[1;;5;;2]] = *x*; *b*
Out[30] = *f*[*x*, x2, *x*, x4, *x*, x6]
In[31] := Clear[*a*, *b*]

3.3 Queries

Let's define an integer number, a rational number, a real (floating point) number, and a complex number.
In[32] := *i* = −1234567890; *r* = −1234567890/987654321;
**   *f* = −1234567890987654321.1234567890987654321; *c* = 1 − 2 ∗ *I*;**
The query AtomQ (Q from Query) returns the symbol True if its argument is an atom and False if it is a composite expression.
In[33] := {AtomQ[*i*], AtomQ[*r*], AtomQ[*c*], AtomQ[*f*[*x*]]}
Out[33] = {True, True, True, False}
The function Head can be applied even to atoms.
In[34] := {Head[*i*], Head[*r*], Head[*c*], Head[*f*]}
Out[34] = {Integer, Rational, Complex, Real}
The function FullForm shows the internal form of an expression with which *Mathematica* operates (to some approximation). For example, a rational number has the head Rational and two arguments—its numerator and denominator.
In[35] := FullForm[*r*]
Out[35]//FullForm =
    Rational[−137174210, 109739369]
A complex number has the head Complex and two arguments—its real and imaginary parts.
In[36] := FullForm[*c*]
Out[36]//FullForm =
    Complex[1, −2]
The internal representation of a floating point number is rather complicated. It contains the mantissa and the exponent and also the number of significant (decimal) digits. In this particular case, there are 37 significant digits.

In[37] := FullForm[*f*]
Out[37]//FullForm =
   − 1.23456789098765432112345678909876543211`37.09151497751671 *^ 18

The query IntegerQ checks if its argument is an integer number.

In[38] := {IntegerQ[*i*], IntegerQ[*r*], IntegerQ[*c*]}
Out[38] = {True, False, False}

The functions Numerator and Denominator extract the parts of a rational number.

In[39] := {Numerator[*r*], Denominator[*r*]}
Out[39] = {−137174210, 109739369}

The functions Re and Im extract the real and imaginary parts of a complex number.

In[40] := {Re[*c*], Im[*c*]}
Out[40] = {1, −2}
In[41] := Clear[*i*, *r*, *f*, *c*]

3.4 Forms of an Expression

FullForm is a very useful function. It shows what *Mathematica* really thinks about an expression. Use it often, and you will learn a lot. For example, the following expression is a sum of 4 terms, one of which is the number -1 multiplied by the symbol z.

In[42] := FullForm[*x* + *y* − *z* − 1]
Out[42]//FullForm =
   Plus[−1, *x*, *y*, Times[−1, *z*]]

And this one is a product of 4 factors, among which are the rational number $2/3$ and the negative power z^{-1}.

In[43] := *a* = 2 ∗ *x* ∗ *y*/(3 ∗ *z*)
Out[43] = $\frac{2xy}{3z}$
In[44] := FullForm[*a*]
Out[44]//FullForm =
   Times[Rational[2, 3], *x*, *y*, Power[*z*, −1]]

Nevertheless, the functions Numerator and Denominator work as expected.

In[45] := {Numerator[*a*], Denominator[*a*]}
Out[45] = {2*xy*, 3*z*}
In[46] := Clear[*a*]

We have already handled lists many times. A list appears to be just the function List with arguments—elements of the list.

In[47] := FullForm[{*x*, *y*, *z*}]
Out[47]//FullForm =
   List[*x*, *y*, *z*]

Any *Mathematica* command can be written as a function with arguments (sometimes, it can also be written in some other way). For example, assignment is the function Set. In order to see this, we'll have to put an assignment inside the function Hold. Otherwise it would be executed immediately, and the function FullForm would receive only the result returned by the assignment—the symbol *x*.

In[48] := FullForm[Hold[*a* = *x*]]

Out[48]//FullForm =

Hold[Set[*a*, *x*]]

Here is a rational expression.

In[49] := *a* = Together[*x*/(*x* + *y*) + *y*/(*x* − *y*)]

Out[49] = $\dfrac{x^2 + y^2}{(x - y)(x + y)}$

Its full form:

In[50] := FullForm[*a*]

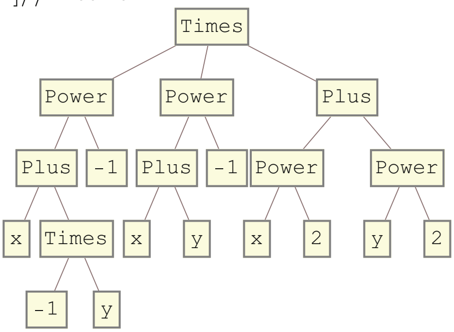
Out[50]//FullForm =

Times[Power[Plus[*x*, Times[−1, *y*]], −1], Power[Plus[*x*, *y*], −1],
Plus[Power[*x*, 2], Power[*y*, 2]]]

And this is the same expression as a tree.

In[51] := TreeForm[*a*]

Out[51]//TreeForm =



In[52] := Clear[*a*]