Lecture 4. Expressions, Part I

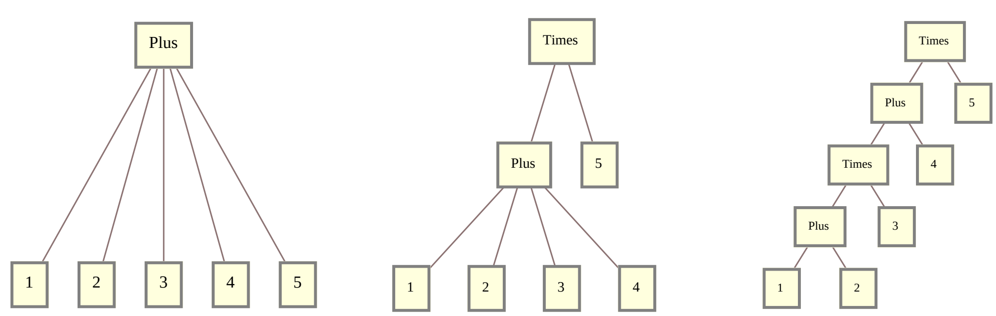
An expression is a written record of a computation. On page 62 of the *Common Core Standards*, a more elaborate account is given, which points out that the computations may start with numbers or with symbols: adding 3 to 10 is a computation, but so is adding 3 to x, if x is a number. Furthermore what is meant by a *computation* might be as simple as adding or multiplying, but taking the square root of a number or taking a trigonometric function of a number or stacking numbers in repeated exponents (as in $x^{x^{x^{\cdot^{\cdot^{\cdot}}}}}$) is also a computation. Beyond this, computations can be strung together or combined with further computations to make complex, multistep computations. When these things are considered, it is clear that we have not yet provided a definition (in the precise mathematical sense) of the concept of an expression, but have only been dancing around with words.

Well, instead of talking about the idea, let us look at some examples. Here are 6 different expressions. Each one uses each of the numbers from 1 to 5 exactly once and records a way of adding and/or multiplying these numbers.

a) $1+2+3+4+5$
b) $(1+2+3+4)5$
c) $((1+2)3+4)5$
d) $(1+2)3+(4+5)$
e) $1+2+3(4+5)$
f) $(1+2)(3)(4)(5)$

Expressions have structure. The structure of an expression can be made clear using a *tree diagram*:

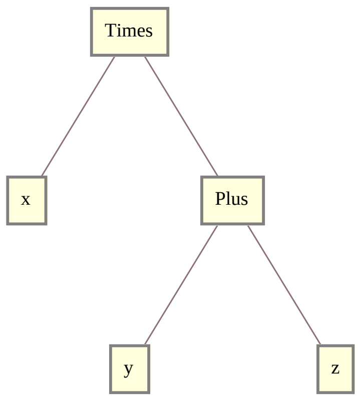
a) $1+2+3+4+5$ *b*) $(1+2+3+4)5$ *c*) $((1+2)3+4)5$

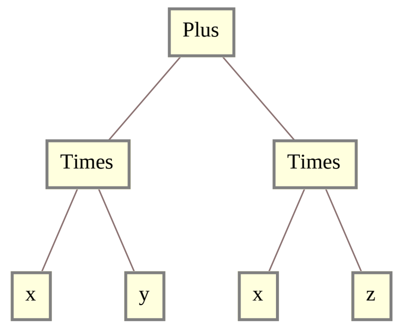


The expressions that we have been talking about do not have any variables in them. We call expressions such as these *numerical expressions* or *arithmetical expressions*. Expressions such at $x+1$ or $a\,x^2+b\,x+c$ that have variables in them are called *algebraic expressions*.

The distributive law is a rule for transforming expressions:

$$x\,(y+z) \qquad = \qquad x\,y + x\,z$$





Problems

1. This problem refers to expressions a)–f) above.
   a) Evaluate the expressions.
   b) Describe in words the computations that correspond to each.
   c) Draw the tree diagrams for c), d) and f).
2. How many different **numbers** can you form, using only addition and multiplication and using each the numbers from 1 to 5 at most once.
3. How many different **expressions** can you form, using only addition and multiplication and using each the numbers from 1 to 5 at most once.
4. Try Problems 2 and 3 with the numbers from 1 to 6. Try with the numbers from 1 to 7.
5. Tom says, "The distributive law tells you that any computation involving additions and multiplications—no matter how long and complex and no matter how many alterations between additions and multiplications—can be done by doing some multiplications first and then doing some additions." Is this correct?

Note. The tree diagrams were made using *Mathematica*. The input for c), for example, was

```
TreeForm[("1"+"2")"3"+"4")"5"]
```

It is necessary to put the numerals in quotation marks because without them, *Mathematica* evaluates the expression. The input

```
TreeForm[((1 + 2) 3 + 4) 5]
```

gives the output `65`. *Mathematica* applies some ordering rules before making the tree, so what it produces will not always have its leaves (the entries at the ends of the branches, at the bottom of the picture) in the order that you put them in.