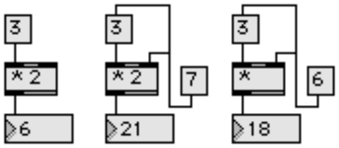
Composition: Electronic Media II
Spring 2009
Math in Max/MSP

Mathematical expressions in Max/MSP are built up from the smallest unit, examples of which are shown below.

1. Here are three expressions:

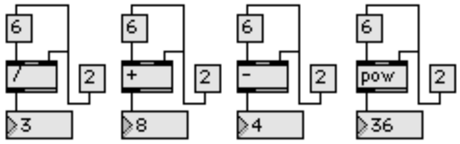


a. The left patch has a message box with "3" on the top. The outlet goes to the inlet of an object box with "* 2" inside. The * is the multiplication object. The number 2 is what Max calls the "argument." I will not use this terminology unless I have to. Click or bang on 3 will produce the number 6 in the number box at the bottom of the patch.

b. The middle patch has a message box, with "7" inside, outputted to two locations. First, it goes to the right inlet of the * object box. This number overrides the number "2" in the * object box. Second, the message "7" box also goes to the inlet of the message "3" box. This means that when the message "7" box is clicked or banged, it in turn sends a bang to the message "3" box. This makes the flow of the expression read: 3 times 7 equal 21.

c. The right patch shows that the * object takes its argument from the message box "6" sent to its right inlet.

2. Other common expressions are:



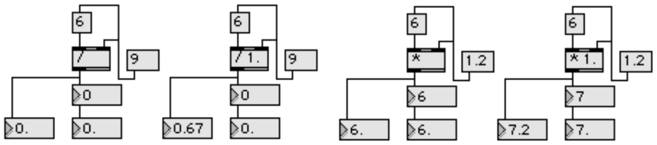
These are: "6 divided by 2 equals 3"
"6 plus 2 equals 8"
"6 minus 2 equals 4"
"6 raised to the 2nd power is 36"

Note: to trigger the expression, click or bang on the message box "2". I recommend using math expressions in this way.

3. Exponents.

a. 3^4 means "3 raised to the 4th power."
b. 3^4 = 3 x 3 x 3 x 3 = 81.
c. In email, you can type 3^4.

4. Decimals.

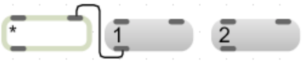


Two things are necessary to show decimals:

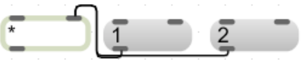
a. One, a float box is needed instead of a number box at the bottom of the patch. As these examples show, the number box strips the decimals.
b. Two, the "/" or "*" object box must include the argument "1." (that is, 1 followed by a period). This is true for all operations, including "+" and "-".

Mod

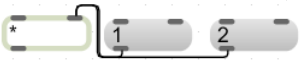
a. Message Box 1 is patched into the right inlet of the multiplier object, as shown below.



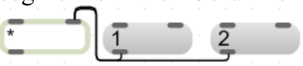
b. Message Box 2 is also patched into the multiplier. This patch cord is superimposed upon the patch cord of Message Box 2. However, the bottom segment of Patch Cord 2 is 1 pixel higher than that of Patch Cord 1, creating a sloppy appearance, as shown below.



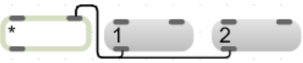
c. The bottom segment of Patch Cord 2 has been shifted 1 pixel down and is now perfectly superimposed upon the bottom segment of Patch Cord 1. However, the vertical segment of Patch Cord 2 is 1 pixel to the right of that of Patch Cord 1, as shown below.



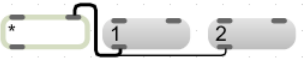
d. The bottom and vertical segments of Patch Cord 2 are perfectly superimposed on those of Patch Cord 1. However, the top segment of Patch Cord 2 is 1 pixel higher than that of Patch Cord 1, as shown below.



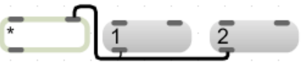
e. Now everything is beautiful, as shown below.



f. To move a segment of Patch Cord 1, click anywhere on the path from Patch Cord 1 to the multiplier. Once Patch Cord 1 is highlighted, the user can grab any segment and move it, as shown below.



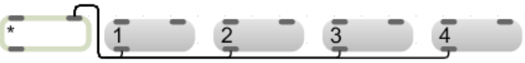
g. To move a segment of Patch Cord 2, click anywhere on the path from the outlet of Message Box 2 to the outlet of Message Box 1. Once Patch Cord 2 is highlighted, the user can grab any segment and move it, as shown below.



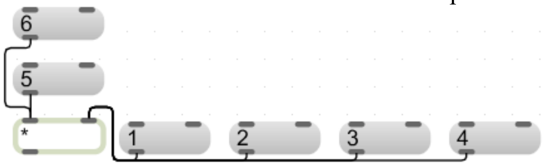
h. Notice that the connection from Message Box 1 to the multiplier should be made before the connection from Message Box 2 to the multiplier. This is because Patch Cord 2 will obscure Patch Cord 1 and make it more difficult to move.

x. Other moves.

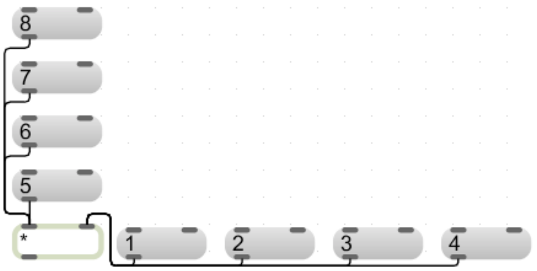
a. A horizontal line of message boxes can be patched into the multiplier in a neat manner, as shown below.



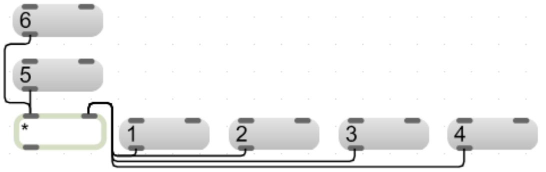
b. A vertical line of message boxes can be patched into the multiplier's left inlet in a neat manner. Notice that the bottom segment of Patch Cord 6 is on the same horizontal plane as the top segments of Patch Cords 1-4, as shown below.



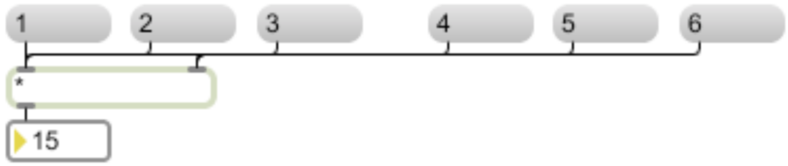
c. Using the same techniques above, Patch Cords 7 and 8 can be superimposed upon Patch Cord 6, as shown below.



d. Sometimes it is preferable to see the individual patch cords, as shown below.

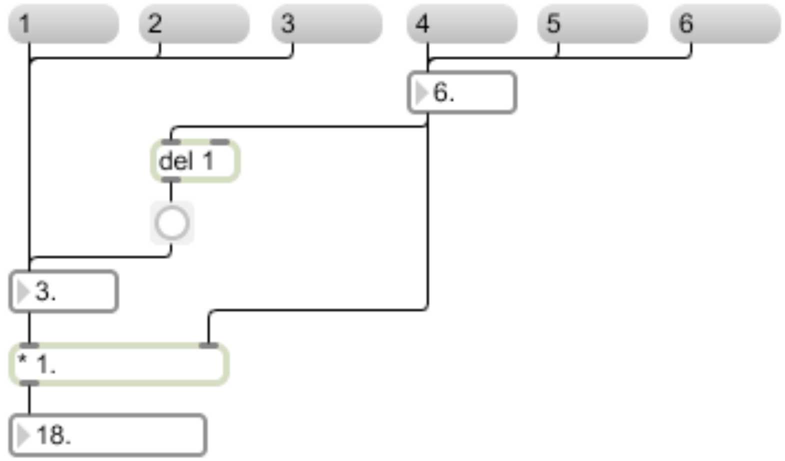


5 * 3 = 15
6 alone = 15
4 alone = 15
2 alone = 8
3 alone = 12



With an operation object, notice that following:

Right input alone does not create a product.
Left input alone always creates a product.
Solution:
If you want an action on the right input to produce a product, you need to:
Send an out


1
2
3
4
5
6
6.
del 1
3.
* 1.
18.