Design ideas for Mathformula2

Roughly speaking, we want to search for areas where activation reflects the syntactic complexity of mathematical expressions.

We want to study the automatic parsing of the formulas, not the associated computation.

The expressions will be briefly presented (200msec) to avoid eye movements. We will use only well known operators to facilitate parsing, limit and the number of operators (maximum 3 or 4: to be tested) and the size of display (10~12 char).

The main issue is to disantangle abstract complexity from visual factors such as number of characters in the formula and presence of parentheses.

To help address the confounding variable 'number of characters', we propose to use variables names of varying length, e.g. greek names (alpha, beta, rho...)

Idea 1. Generate expressions such as in the following table. The analysis will use a main hrf for all stimuli, with parametric modulators: number of unary operators, number of binary operators, number of parentheses and number of characters. Issue: can we sufficiently distangle the 4 variables?

Expression	# unary op	# binary op		# parenth	# char
x+y	0	1		0	3
(x+y)	0	1		2	5
x*(y+z)	0	2		2	7
x*y	0	1		0	3
x*y+z	0	2		0	5
sin(x)	1	0		2	6
x-(y+z)	0	2		2	7
alpha+beta	0	1		0	10
alpha*(x+y)	0	2		2	11
ln(x)	1	0		2	5
arctan(x)	1	0		2	8
sin(x)+cos(y)	2	1		4	13
sin(alpha)	1	0		2	10
ln(x)+exp(y)	2	1		4	12
x*ln(y)	1	1		2	7
x+y*(t+z)	0	4		2	9

Idea 2: Use a limited numbers of trees, so that we can estimate the reponse to each

type of trees, and test various models of composition. For example:

Set 1: 1 unary op	sin(x), cos(alpha), ln(rho), atan(mu)
Set 2: 1 binary op	x+y, x*alpha, x-z
Set 3: 2 unary op	ln(sin(x)), cos(atan(t))
Set 4: 2 binary op	alpha+y*z x*(y-z)
Set 5: 1 binary 1 un	x+ln(y)
Set 6: 1 binary 1 un	ln(x+y)
Set 7: 1 binary 2 un	sin(x)+ln(y)
Set 8: 2 binary 1 un	alpha+x*log(x)

From the activation patterns of Set1, Set2, can we “explain” the actaivtioan patterns of Set 3 to 8?