Methods

Assume we have four positive integers a,b,c,d, the task is to use `(,),+-*/floor` to get 24. Every number should be used and used only once.

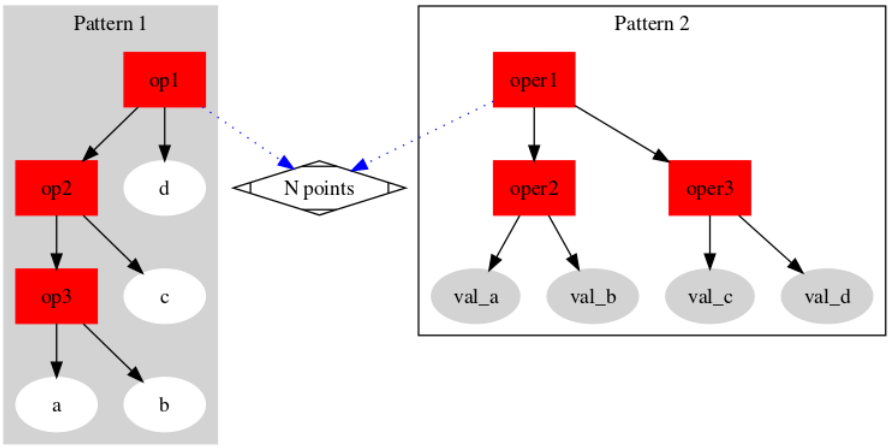
R routine

R is likely to be a **functional program**, which could use trich like `lisp` or `scheme`. I don't mean to write something like

```
(+ (+ a b) (* c d))
```

Here it is actually: `(a+b)+(c*d)`

But we could use similar structure shown in



.

Check src.

This is a much interesting version(like scheme macro) src2

I used the same data structure

```
c("Patten","Op1","Op2","Op3",letters[1:4])
```

, which could be used to generate/parsing

```
# pat 1
op1(op2(a,b),op3(c,d))
# par 2
"op3"("op2"("op1"(a,b),c),d)
#i.e.
((a op1 b) op2 c) op3 d
```

Appendix

dots file

```
digraph G1{
compound=true;
subgraph clusterPattern1{
node[style=filled color=white];
style=filled;
color=lightgrey;
op1 -> op2 -> op3 -> a;
op1 -> d;
op2 -> c;
op3 -> b;
op1[shape="box",style=filled,color=red];
op2[shape="box",style=filled,color=red];
op3[shape="box",style=filled,color=red];
label ="Pattern 1";
}

subgraph clusterPattern2{
node[style=filled color=lightgrey];
oper1 -> {oper2 oper3};
oper2 -> {val_a val_b}
oper3 -> {val_c val_d};
color = black;
oper1[shape="box",style=filled,color=red];
oper2[shape="box",style=filled,color=red];
oper3[shape="box",style=filled,color=red];
label="Pattern 2"
}

{op1 oper1}->Root[style=dotted color=blue];
Root[shape=Mdiamond,label="N points"];


}
```