

The patterns are we matching against is float modules types, functional operators and pytorch operators in reverse order:

```
operator = module_type | functional | torch op | native op | MatchAllNode
Pattern = (operator, Pattern, Pattern, ...) | operator
```

where the first item for Pattern is the operator we want to match, and the rest are the patterns for the arguments of the operator. For example, pattern (nn.ReLU, (operator.add, MatchAllNode, (nn.BatchNorm2d, nn.Conv2d))) would match the following graph:

```

tensor_1          tensor_2
 |                |
 |(MatchAllNode)  | nn.Conv2d
 |                |
 |                | nn.BatchNorm2d
 \                /
  -- operator.add --
      |
    nn.ReLU
```

we'll match the last node as the anchor point of the match, and we can retrieve the whole graph by tracing back from the node, e.g. in the example above, we matched nn.ReLU node, then node.args[0] is the operator.add node.