phylanx

November 8, 2017

```
In [1]: from sympy import symbols, sin, srepr
        x, y, z = symbols('x y z')
        z = x * y
        sin_z = sin(z)
        expr = sin_z / 2
        print(expr)
        print(srepr(expr))
```

```
sin(x*y)/2
Mul(Rational(1, 2), sin(Mul(Symbol('x'), Symbol('y'))))
```

```
In [2]: from sympy import preorder_traversal
        for arg in preorder_traversal(expr):
            print(arg)
```

```
sin(x*y)/2
1/2
sin(x*y)
x*y
x
y
```

```
In [3]: from sympy import postorder_traversal
        for arg in postorder_traversal(expr):
            print(arg)
```

```
1/2
x
y
x*y
sin(x*y)
sin(x*y)/2
```

```
In [4]: from sympy import MatrixSymbol, Matrix

        A = MatrixSymbol('A', 4, 4)
        B = MatrixSymbol('B', 4, 4)
```

```
In [5]: print(A)
```

```
A
```

```
In [6]: print(srepr(A))
```

```
MatrixSymbol('A', Integer(4), Integer(4))
```

```
In [7]: C = A + B
        print(C)
        print(srepr(C))
```

```
A + B
MatAdd(MatrixSymbol('A', Integer(4), Integer(4)), MatrixSymbol('B', Integer(4), Integer(4)))
```

```
In [8]: A1 = A[:2, :]
        A2 = A[2:, :]
        print(A1)
        print(srepr(A1))
```

```
A[:2, :4]
MatrixSlice(MatrixSymbol('A', Integer(4), Integer(4)), Tuple(Integer(0), Integer(2), Integer(1))
```

```
In [9]: A_ = Matrix(A1).row_join(Matrix(A2))
        print(A_)
```

```
Matrix([[A[0, 0], A[0, 1], A[0, 2], A[0, 3], A[2, 0], A[2, 1], A[2, 2], A[2, 3]], [A[1, 0], A[1,
```