```
#
#
#
#
#
#
#
```

expand

```
from casadi import *
import casadi as c
```

We construct a simple MX expression

```
x = MX.sym("x",2,2)
y = MX.sym("y",2,1)

z = mtimes(x,y)
```

Let's construct an MXfunction

```
f = Function("f", [x,y],[z])
```

We expand the MX expression into an SX expression

```
fSX = f.expand('fSX')

print "Expanded expression = ", fSX
```

```
Expanded expression =   Number of inputs: 2
  Input 0 ("i0"): 2-by-2 (dense)
  Input 1 ("i1"): 2-by-1 (dense)
 Number of outputs: 1
  Output 0 ("o0"): 2-by-1 (dense)
@0 = input[0][0];
@1 = input[1][0];
@0 = (@0*@1);
@2 = input[0][2];
@3 = input[1][1];
@2 = (@2*@3);
@0 = (@0+@2);
output[0][0] = @0;
@0 = input[0][1];
@0 = (@0*@1);
@1 = input[0][3];
@1 = (@1*@3);
@0 = (@0+@1);
output[0][1] = @0;
```

Limitations

Not all MX graphs can be expanded. Here is an example of a situation where it will not work.

```
linear_solver = linsol("linear_solver", "csparse", x.sparsity(), 1)
g = linear_solver.linsol_solve(x, y)
G = Function("G", [x,y], [g])
```

This function cannot be expanded.

```
try:
  G.expand('G_sx')
except Exception as e:
  print e
```

```
 on line 436 of file "/home/travis/build/casadi/binaries/casadi/casadi/
    core/function/linsol.cpp"
Linsol::eval_sxLinsol not defined for class N6casadi16CsparseInterfaceE
```