Symbolic substitution

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

```
from casadi import *
```

Let’s build a trivial symbolic SX graph

```
x = SX.sym("x")
y = SX.sym("y")
z_= x*y
z = z_+x
print type(z), z
```

```
<class 'casadi.casadi.SX'> ((x*y)+x)
```

We need SXFuncion to manipulate the SX graph

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

We can substitute a leaf in the graph

```
w = SX.sym("w")
q = f(vertcat(w,y))
```

f.eval() returns a tuple with all outputs, we selected the first

```
print type(q), q
```

```
<class 'casadi.casadi.SX'> ((w*y)+w)
```

Note how q is now an SX
We can take a shortcut via substitute:

```
q = substitute(z,x,w)
print type(q), q
```

```
<class 'casadi.casadi.SX'> ((w*y)+w)
```

Note that substitution of non-symbolic SX nodes is not permitted:
This is actually a restriction of Function: