

# n\_nodes

April 18, 2023

This file is part of CasADi.

CasADi -- A symbolic framework for dynamic optimization.  
Copyright (C) 2010–2023 Joel Andersson, Joris Gillis, Moritz Diehl,  
KU Leuven. All rights reserved.  
Copyright (C) 2011–2014 Greg Horn

CasADi is free software; you can redistribute it and/or  
modify it under the terms of the GNU Lesser General Public  
License as published by the Free Software Foundation; either  
version 3 of the License, or (at your option) any later version.

CasADi is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public  
License along with CasADi; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

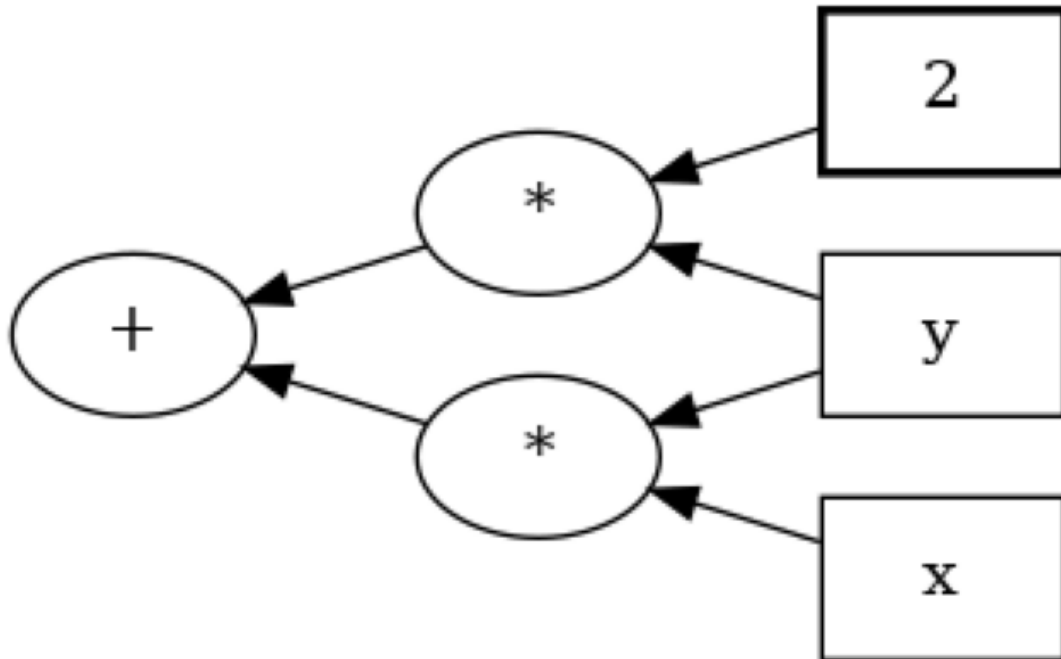
## 1 n\_nodes

```
[1]: from casadi import *  
     from casadi.tools import *
```

Let's build a trivial symbolic SX graph

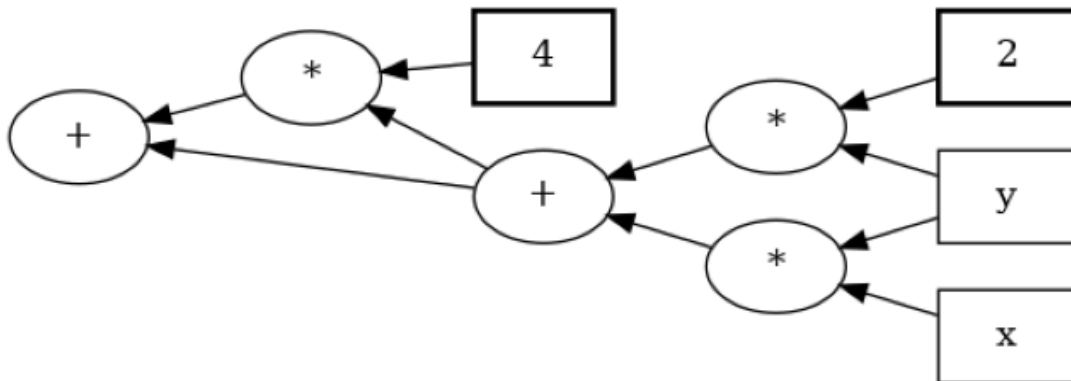
```
[2]: x = SX.sym("x")  
     y = SX.sym("y")  
     z = x*y+2*y  
     print(n_nodes(z), " nodes in ", z)  
     dotdraw(z)
```

```
6 nodes in ((x*y)+(2*y))
```



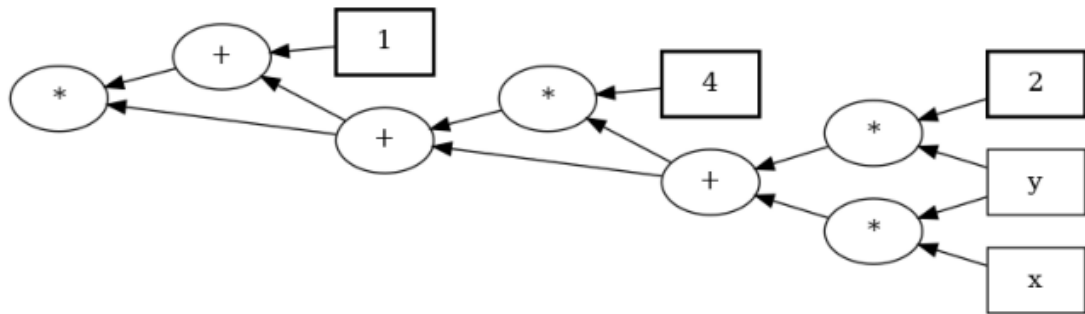
```
[3]: z += 4*z
print(n_nodes(z), " nodes in ", z)
dotdraw(z)
```

9 nodes in @1=((x\*y)+(2\*y)), (@1+(4\*@1))



```
[4]: z *= z+1
print(n_nodes(z), " nodes in ", z)
dotdraw(z)
```

```
12 nodes in @1=((x*y)+(2*y)), @2=(@1+(4*@1)), (@2*(@2+1))
```



[ ]: