Hands-on CasADi course on optimal control, Yacoda

Exercise: CasADi basics

For these exercises, assume the following objects are defined:

```
from casadi import *
x = MX.sym("x");     y = MX.sym("y")      # scalar symbol
p = MX.sym("p",2);   A = MX.sym("A",4,5)  # vector and matrix symbol
f = Function("f",[x,y],[sin(x)*y])
g = Function("g",[x],[sqrt(x),x**2])
h = Function("h",[x],[sin(x)*y])
F = lambda x,y: sin(x)*y
```

1 What's the type?

For each code snippet, encircle the character corresponding to the most appropriate data-type. When in doubt, enter the code snippet in Python and use `type(.)`.

Nr	Code	MX (constant)	MX (other)	DM	Function	non-CasADi type
1	`5`	d	i	u	h	e
2	`5+x`	s	l	g	z	m
3	`x-x`	e	d	b	k	a
4	`x==x`	c	r	z	u	v
5	`x==y`	m	t	s	j	f
6	`A*x`	a	r	v	w	c
7	`A*0`	o	t	e	a	f
8	`gradient(sin(x),x)`	h	e	p	m	g
9	`gradient(x,x)`	n	x	m	i	b
10	`p[1]`	q	c	j	m	o
11	`A[:,0]`	z	e	f	m	u
12	`f(1,2)`	m	b	p	j	g
13	`f(1,y)`	j	h	l	r	q
14	`g(9)`	m	b	e	f	a
15	`g(9)[0]`	q	t	l	n	s
16	`g(x)`	m	a	q	r	o
17	`g(x)[0]`	b	g	c	y	o
18	`g(x)[0][0]`	i	r	o	j	u
19	`h(x)`	e	a	y	h	u
20	`f.expand()`	d	b	a	p	w
21	`F`	e	f	p	n	h
22	`F(x,y)`	g	e	b	a	o
23	`F(1,2)`	c	s	a	h	r

Solution: electro-encephalographer

2 Correct the error

Goal 1: Compose a Function $x \mapsto \sin(x)$
Tried: `Function('m',{x},{sin(x)})`

```
NotImplementedError: Wrong number or type of arguments
for overloaded function 'new_Function'.
  Possible prototypes are:
    Function(str,[MX],[MX])
  You have: '(str,set,set)'
```

Correction: .....

Goal 2: Get first element of p vector
Tried: `p(1)`

```
TypeError: 'MX' object is not callable
```

Correction: .....

Goal 3: Get last element of p vector
Tried: `p[2]`

```
Out of bounds error. Got elements in range [2,2],
which is outside the range [-2,2).
```

Correction: .....

Goal 4: Concatenate two symbolic expressions
Tried: `M=np.zeros((2,1));M[0]=x;M[1]=y;`

```
Implicit conversion of symbolic CasADi type to numeric matrix not supported.
This may occur when you pass a CasADi object to a numpy function.
Use an equivalent CasADi function instead of that numpy function.
```

Correction: .....

Goal 5: Call Function f numerically for x=1,y=2
Tried: `f[1,2]`

```
'Function' object is not subscriptable
```

Correction: .....

Goal 6: Call Function f numerically with elements of p
Tried: `f(p)`

```
Incorrect number of inputs: Expected 2, got 1
```

Correction: .....

Goal 7: Create a Function that returns the square of `f`'s output
Tried: `f**2`

```
unsupported operand type(s) for ** or pow(): 'Function' and 'int'
```

Correction: .....

Goal 8: Evaluate h for $x = 5$
Tried: `h(5)`

```
Cannot evaluate "f:(i0)->(o0) MXFunction" since variables [y] are free
```

Correction: .....

Goal 9: Compose a Function $x \mapsto 5$
Tried: `Function('m',[5],[x])`

```
Xfunction input arguments must be purely symbolic.
Argument 0(i0) is not symbolic.
```

Correction: .....

Goal 10: Compose a Function $x \mapsto \sin(x)$
Tried: `Function('m',[x],[sin])`

```
NotImplementedError: Wrong number or type of arguments
for overloaded function 'new_Function'.
  Possible prototypes are:
    Function(str,[MX],[MX])
  You have: '(str,[MX],[function])
```

Correction: .....

Goal 11: Compose a Function with two input arguments where the second one is ignored
Tried: `Function('m',[x,x],[x])`

```
The input expressions are not independent:
0: x
1: x.
```

Correction: .....

Solutions:

1. `Function('m',[x],[sin(x)])`
2. `p[0]`
3. `p[1]`
4. `M=MX(np.zeros((2,1)));M[0]=x;M[1]=y`
   `or M=MX.zeros(2,1);M[0]=x;M[1]=y`
   `or M = vertcat(x,y) (preferred)`
5. `f(1,2)`
6. `f(p[0],p[1])`
7. `Function("m",[x,y],[f(x,y)**2])`
8. `h(MX(5))`
9. `Function('m',[x],[5])`
10. `Function('m',[x],[sin(x)])`
11. `Function('m',[x,MX()],[x])`

3 What's the type (part 2)?

CasADi Functions can also be called with keyword-value arguments...

```
f2 = Function("f2",[x,y],[sin(x)+y],["x","y"],["z"])
g2 = Function("g2",[x],[sqrt(x),x**2],["x"],["p0","p1"])
```

Nr	Code	MX (constant)	MX (other)	DM	Function	non-CasADi type	Error
1	`f2(3)`	k	o	i	a	t	d
2	`f2(3,4)`	f	z	i	t	h	d
3	`f2(y=4,x=3)`	p	o	q	h	s	y
4	`f2(y=4,x=3)["z"]`	x	o	p	g	a	j
5	`f2(x=x,y=4)["z"]`	c	r	q	u	d	z
6	`f2(x=3)`	g	n	p	u	o	i
7	`f(x=3)`	r	n	u	k	f	p
8	`f(3)`	d	g	n	q	l	o
9	`f2()`	m	n	v	c	r	j
10	`f2()["z"]`	r	x	t	y	a	g
11	`g2(x)`	o	u	d	n	i	m
12	`g2(x=3)`	b	m	a	v	o	q
13	`g(3)`	u	i	z	c	n	q
14	`g(3)[0]`	b	r	a	y	p	j
15	`g2(x=3)[0]`	l	j	p	i	a	t
16	`g2([])`	y	n	h	q	e	p

Solution: disproportionate