

Domáce zadanie na druhé cvičenie

22. novembra 2023

1 Úvod

Nasledujúce zadanie reprezentuje racionálne¹ funkcie prefixovo takto

funkcia	reprezentácia
$f(x) = x$	'x'
$f(x) = 1$	1
$f(x) = 2x$	['*' 2 'x']
$f(x, y) = \frac{x+2y}{3+\frac{2.5}{x}}$	['/', ['+', 'x', ['*', 2, 'y']], ['+', 3, ['/', 2.5, 'x']]]

Tomuto budeme hovoriť prefixová reprezentácia. Uvedomte si, že v tejto reprezentácii je konštantná funkcia všade rovná 5.2 reprezentovaná ako číslo 5.2 a funkcia všade rovná x reprezentovaná ako reťazec 'x'.

Zrejme spracovanie takýchto prefixových reprezentácií bude rekurzívne.

2 Derivujte

Napište funkciu, `myderive(f, var)`, ktorá vráti prefixovú reprezentáciu parciálnej derivácie funkcie `f` danej prefixovou reprezentáciou.

¹Nemusíte implementovať umocňovanie na konštantu, stačí ['+', '-', '*', '/'].

```

>>> myderive(1,"x")
0
>>> myderive("y","x")
0
>>> myderive("x","x")
1
>>> myderive("y","x")
0
>>> myderive(["-",2,"x"],"x")
['-', 0, 1]
>>> myderive(["*",2,"x"],"x")
['+', ['*', 0, 'x'], ['*', 2, 1]]
>>> myderive(["*", "x", "x"], "x")
['+', ['*', 1, 'x'], ['*', 'x', 1]]
>>> myderive(["*", "x", "x"], "y")
['+', ['*', 0, 'x'], ['*', 'x', 0]]
>>> myderive(["*", ["-", "x", 1], "x"], "x")
['+', ['*', ['-', 1, 0], 'x'], ['*', ['-', 'x', 1], 1]]
>>> myderive(["+", "x", "x"], "x")
['+', 1, 1]
>>> myderive(["+", "y", "x"], "x")
['+', 0, 1]
>>> myderive(["/", "x", "y"], "y")
['/', ['-', ['*', 0, 'y'], ['*', 'x', 1]], ['*', 'y', 'y']]

```

3 Pomôcky

- Samozrejme, `myderive` bude rekurzívna funkcia. Základné prípady sú, keď je `f` číslo alebo reťazec. Medziným musí teda vaša funkcia zistiť typ parametra `f` a na základe toho sa rozhodnúť, čo robiť. Robí sa to takto:

```

>>> type(1)==int
True
>>> type(1)==list
False
>>> print(type('x'))
<type 'str'>
>>> type('x')==str

```

```

True
>>> type('x')==list
False
>>> type(['*',2,'x'])==list
True
>>> isinstance(2,int)
True
>>> type(2.5)==int
False
>>> type(2.5)==float
True
>>> type(1)==type(2)
True
>>> type(1)==type("x")
False
>>>

```

- Uvedomte si, že stačí implementovať známe pravidlá o derivácii konštanty, premennej a aritmetických operácií nad funkciami.
- Je dobré si uvedomiť, že napríklad pravidlo o násobení konštantou nemusíte implementovať, pretože vyplýva z pravidla o derivácii konštanty a pravidla o derivácii súčinu funkcií.
- Skôr ako začnete komplikovane pomocou pomocných premenných konštruovať zoznamy cez `extend` a `append`, si uvedomte, že v Pythone môžete bez problémov používať výrazy ako `['*',niečo,niečo_iné]` priamo v kóde.