Trabajo Bioinformática*

Ignacio Amat Hernández Ángela LLopis Hernández Estrella Ballester Hoyo

25 de abril de 2020

*Grado en Ingeniería Biomédica, Escuela Técnica Superior de Ingenieros Industriales, Valencia, España.

Hemos implementado las trazas de manera que son perfectamente idénticas en Python y en C.

```
                                  Mutation    i   z   k   l
Deletion                          [l]         1  -1   0   6
Insertion                         [g]         2  -1   1   2
Substitution                [g] -> [l]        1  -1   1   2
Insertion                         [l]         2  -1   3   3
Match                             [l]         1   0   3   3
 Deletion                         [o]         0  -1   3   3
 Insertion                        [o]         1  -1   5   5
 Match                            [o]         0   0   5   5
  Deletion                        [g]        -1  -1   5   5
  Insertion                       [g]         0  -1   1   1
  Match                           [g]        -1   0   1   1
———————————————————————— {1,1} ——————————————————
Insertion                         [o]         2  -1   4   6
Substitution                [o] -> [l]        1  -1   4   6
```

Cuadro 1: Traza de InexRecur con X = "googol$", W = "gol", z = 0 en C y Python

```
|      INEXRECUR   -    by  XAVI  GABRI  AITANA  ALFREDO       |
-     D        [ l ]        1     -1     0     6
-     I        [ g ]        2     -1     1     2
-     S     [ g -> l ]    1     -1     1     2
-     I        [ l ]        2     -1     3     3
-     M        [ l ]        1     0     3     3
-     -     D        [ o ]        0     -1     3     3
-     -     I        [ o ]        1     -1     5     5
-     -     M        [ o ]        0     0     5     5
-     -     -     D        [ g ]        -1     -1     5     5
-     -     -     I        [ g ]        0     -1     1     1
-     -     -     M        [ g ]        -1     0     1     1
?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?->   [ c(1, 1) ]
-     I        [ o ]        2     -1     4     6
-     S     [ o -> l ]    1     -1     4     6
```

Cuadro 2: Traza de InexRecur con X = "googol$", W = "gol", z = 0 en R

1		Mutation	i	z	k	l
2	Deletion	[g]	2	−1	0	6
3	Insertion	[g]	3	−1	1	2
4	Match	[g]	2	0	1	2
5	Deletion	[o]	1	−1	1	2
6	Insertion	[o]	2	−1	4	4
7	Match	[o]	1	0	4	4
8	Deletion	[o]	0	−1	4	4
9	Insertion	[o]	1	−1	6	6
10	Match	[o]	0	0	6	6
11	Insertion	[l]	3	−1	3	3
12	Substitution	[l] −> [g]	2	−1	3	3
13	Insertion	[o]	3	−1	4	6
14	Substitution	[o] −> [g]	2	−1	4	6

Cuadro 3: Traza de InexRecur con X = “googol$”, W = “goog”, z = 0 en C y Python

1	|	INEXRECUR − by XAVI GABRI AITANA ALFREDO										|
2	−	D	[g]			2	−1	0	6			
3	−	I	[g]			3	−1	1	2			
4	−	M	[g]			2	0	1	2			
5	−	−	D	[o]		1	−1	1	2			
6	−	−	I	[o]		2	−1	4	4			
7	−	−	M	[o]		1	0	4	4			
8	−	−	−	D	[o]	0	−1	4	4			
9	−	−	−	I	[o]	1	−1	6	6			
10	−	−	−	M	[o]	0	0	6	6			
11	−	I	[l]			3	−1	3	3			
12	−	S	[l −> g]			2	−1	3	3			
13	−	I	[o]			3	−1	4	6			
14	−	S	[o −> g]			2	−1	4	6			

Cuadro 4: Traza de InexRecur con X = “googol$”, W = “goog”, z = 0 en R

	Mutation	i	z	k	l
Deletion	[l]	2	0	1	6
Deletion	[o]	1	−1	1	6
Insertion	[g]	2	−1	1	2
Substitution	[g] −> [o]	1	−1	1	2
Insertion	[o]	2	−1	4	6
Match	[o]	1	0	4	6
Deletion	[o]	0	−1	4	6
Insertion	[g]	1	−1	1	2
Substitution	[g] −> [o]	0	−1	1	2
Insertion	[o]	1	−1	6	6
Match	[o]	0	0	6	6
Deletion	[g]	−1	−1	6	6
Insertion	[g]	0	−1	2	2
Match	[g]	−1	0	2	2
———	{2,2}	———	———	———	———
Insertion	[g]	3	0	1	2
Substitution	[g] −> [l]	2	0	1	2
Deletion	[o]	1	−1	1	2
Insertion	[o]	2	−1	4	4
Match	[o]	1	0	4	4
Deletion	[o]	0	−1	4	4
Insertion	[o]	1	−1	6	6
Match	[o]	0	0	6	6
Deletion	[g]	−1	−1	6	6
Insertion	[g]	0	−1	2	2
Match	[g]	−1	0	2	2
———	{2,2}	———	———	———	———
Insertion	[o]	3	0	4	6
Substitution	[o] −> [l]	2	0	4	6
Deletion	[o]	1	−1	4	6
Insertion	[g]	2	−1	1	2
Substitution	[g] −> [o]	1	−1	1	2
Insertion	[o]	2	−1	6	6
Match	[o]	1	0	6	6
Deletion	[o]	0	−1	6	6
Insertion	[g]	1	−1	2	2
Substitution	[g] −> [o]	0	−1	2	2

Cuadro 5: Traza de InexRecur con X = “googol$”, W = “gool”, z = 1 en C y Python

```
1  |      INEXRECUR  -  by XAVI GABRI AITANA ALFREDO      |
2  -    D       [ l ]        2    0    1    6
3  -    -    D       [ o ]        1   -1    1    6
4  -    -    I       [ g ]        2   -1    1    2
5  -    -    S    [ g -> o ]     1   -1    1    2
6  -    -    I       [ o ]        2   -1    4    6
7  -    -    M       [ o ]        1    0    4    6
8  -    -    -    D       [ o ]        0   -1    4    6
9  -    -    -    I       [ g ]        1   -1    1    2
10 -    -    -    S    [ g -> o ]     0   -1    1    2
11 -    -    -    I       [ o ]        1   -1    6    6
12 -    -    -    M       [ o ]        0    0    6    6
13 -    -    -    -    D       [ g ]        -1   -1    6    6
14 -    -    -    -    I       [ g ]        0   -1    2    2
15 -    -    -    -    M       [ g ]        -1    0    2    2
16 ?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-> [ c(2, 2) ]
17 -    I       [ g ]        3    0    1    2
18 -    S    [ g -> l ]     2    0    1    2
19 -    -    D       [ o ]        1   -1    1    2
20 -    -    I       [ o ]        2   -1    4    4
21 -    -    M       [ o ]        1    0    4    4
22 -    -    -    D       [ o ]        0   -1    4    4
23 -    -    -    I       [ o ]        1   -1    6    6
24 -    -    -    M       [ o ]        0    0    6    6
25 -    -    -    -    D       [ g ]        -1   -1    6    6
26 -    -    -    -    I       [ g ]        0   -1    2    2
27 -    -    -    -    M       [ g ]        -1    0    2    2
28 ?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-?-> [ c(2, 2) ]
29 -    I       [ o ]        3    0    4    6
30 -    S    [ o -> l ]     2    0    4    6
31 -    -    D       [ o ]        1   -1    4    6
32 -    -    I       [ g ]        2   -1    1    2
33 -    -    S    [ g -> o ]     1   -1    1    2
34 -    -    I       [ o ]        2   -1    6    6
35 -    -    M       [ o ]        1    0    6    6
36 -    -    -    D       [ o ]        0   -1    6    6
37 -    -    -    I       [ g ]        1   -1    2    2
38 -    -    -    S    [ g -> o ]     0   -1    2    2
```

Cuadro 6: Traza de InexRecur con X = "googol$", W = "gool", z = 1 en R

```
inexrecur_time.c
12.34 µs from 10000 iterations.

inexrecur_time.py
real      sys     user
268.89µs  0.37µs  268.16µs

inexrecur_time.R
user      system    elapsed
5422µs    18µs      5443µs
```

Cuadro 7: Tiempos de "CPU".

Ejecutando desde la línea de comandos como scripts usando `#!/bin/env Rscript` y `#!/bin/env Python`.

```
bench ./inexrecur_clean ./inexrecur_clean.py ./inexrecur_clean.R
benchmarking bench/./inexrecur_clean
time                 4.524 ms    (4.496 ms .. 4.551 ms)
                     0.999 R²    (0.999 R² .. 1.000 R²)
mean                 4.522 ms    (4.499 ms .. 4.559 ms)
std dev              89.83 µs    (58.07 µs .. 133.1 µs)

benchmarking bench/./inexrecur_clean.py
time                 39.26 ms   (39.12 ms .. 39.40 ms)
                     1.000 R²   (1.000 R² .. 1.000 R²)
mean                 39.30 ms   (39.18 ms .. 39.45 ms)
std dev              266.5 µs   (177.4 µs .. 388.7 µs)

benchmarking bench/./inexrecur_clean.R
time                 278.5 ms   (273.6 ms .. 285.0 ms)
                     1.000 R²   (1.000 R² .. 1.000 R²)
mean                 280.6 ms   (279.1 ms .. 281.9 ms)
std dev              1.714 ms   (1.027 ms .. 2.517 ms)
variance introduced by outliers: 16% (moderately inflated)
```

Cuadro 8: Tiempo de "pared" según la utilidad *bench*.