Trabajo Bioinformática[*]

Ignacio Amat Hernández Ángela LLopis Hernández Estrella Ballester Hoyo

27 de abril de 2020

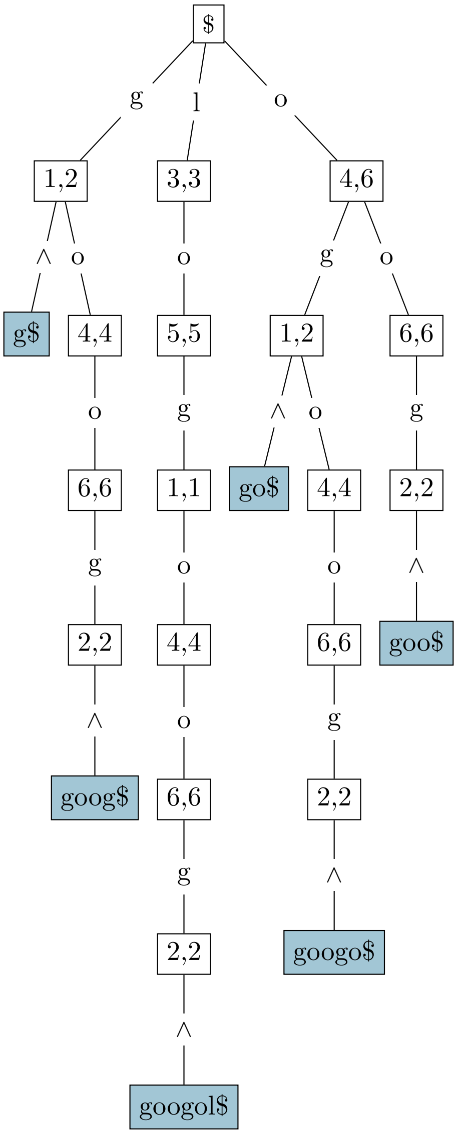

Figura 1: Árbol de prefijos con intervalos SA de la palabra “googol$”.

[*] 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

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

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

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

```
|      INEXRECUR   –   by  XAVI  GABRI  AITANA  ALFREDO       |
–    D        [ l ]          2    0    1    6
–    –    D        [ o ]          1    –1    1    6
–    –    I        [ g ]          2    –1    1    2
–    –    S     [ g –> o ]     1    –1    1    2
–    –    I        [ o ]          2    –1    4    6
–    –    M        [ o ]          1    0    4    6
–    –    –    D        [ o ]          0    –1    4    6
–    –    –    I        [ g ]          1    –1    1    2
–    –    –    S     [ g –> o ]     0    –1    1    2
–    –    –    I        [ o ]          1    –1    6    6
–    –    –    M        [ o ]          0    0    6    6
–    –    –    –    D        [ g ]          –1    –1    6    6
–    –    –    –    I        [ g ]          0    –1    2    2
–    –    –    –    M        [ g ]          –1    0    2    2
?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–>  [ c(2,  2) ]
–    I        [ g ]          3    0    1    2
–    S     [ g –> l ]     2    0    1    2
–    –    D        [ o ]          1    –1    1    2
–    –    I        [ o ]          2    –1    4    4
–    –    M        [ o ]          1    0    4    4
–    –    –    D        [ o ]          0    –1    4    4
–    –    –    I        [ o ]          1    –1    6    6
–    –    –    M        [ o ]          0    0    6    6
–    –    –    –    D        [ g ]          –1    –1    6    6
–    –    –    –    I        [ g ]          0    –1    2    2
–    –    –    –    M        [ g ]          –1    0    2    2
?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–?–>  [ c(2,  2) ]
–    I        [ o ]          3    0    4    6
–    S     [ o –> l ]     2    0    4    6
–    –    D        [ o ]          1    –1    4    6
–    –    I        [ g ]          2    –1    1    2
–    –    S     [ g –> o ]     1    –1    1    2
–    –    I        [ o ]          2    –1    6    6
–    –    M        [ o ]          1    0    6    6
–    –    –    D        [ o ]          0    –1    6    6
–    –    –    I        [ g ]          1    –1    2    2
–    –    –    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 R^2    (0.999 R^2 .. 1.000 R^2)
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 R^2   (1.000 R^2 .. 1.000 R^2)
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 R^2   (1.000 R^2 .. 1.000 R^2)
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*.

```
inexrecur_clean.c
==81469==     in use at exit: 18,588 bytes in 164 blocks
==81469==   total heap usage: 191 allocs, 27 frees, 24,894 bytes allocated

inexrecur_mem.py
6,631,424 bytes

inexrecur_mem.R
4,932,584 bytes
```

Cuadro 9: Memoria usada medido desde el script.

```
time ./inexrecur_clean
(5,5)
(1,1)
./inexrecur_clean
0.00s  user 0.00s system 44% cpu 0.004 total
max memory:                    676 kB

time ./inexrecur_clean.py
(1, 1)
(5, 5)
./inexrecur_clean.py
0.03s  user 0.01s system 82% cpu 0.039 total
max memory:                    6272 kB

time ./inexrecur_clean.R
[[1]]
[1] 5 5

[[2]]
[1] 1 1

./inexrecur_clean.R
0.23s  user 0.04s system 97% cpu 0.277 total
max memory:                    67476 kB
```

Cuadro 10: Memoria usada medido desde fuera del script.

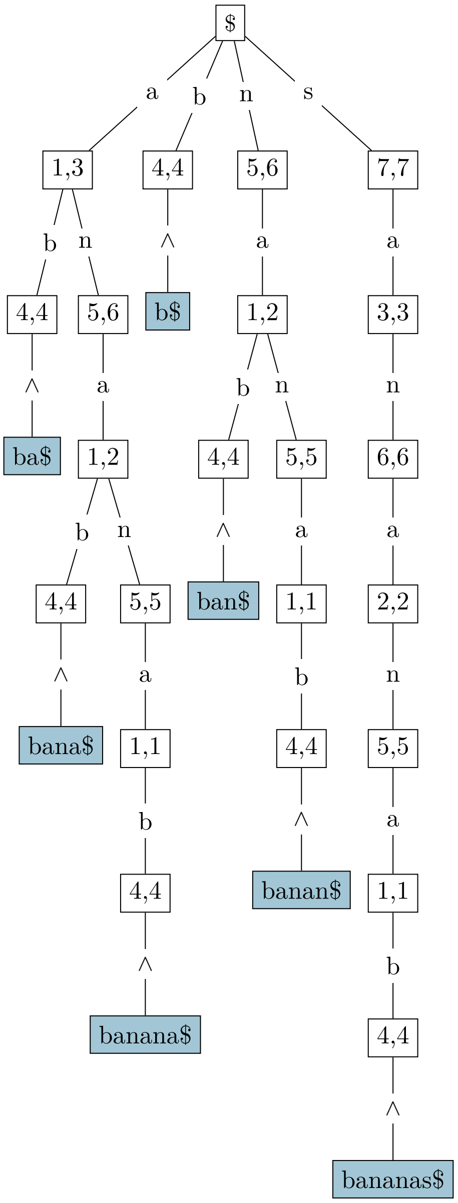

Figura 2: Árbol de prefijos con intervalos SA de la palabra “bananas$”.

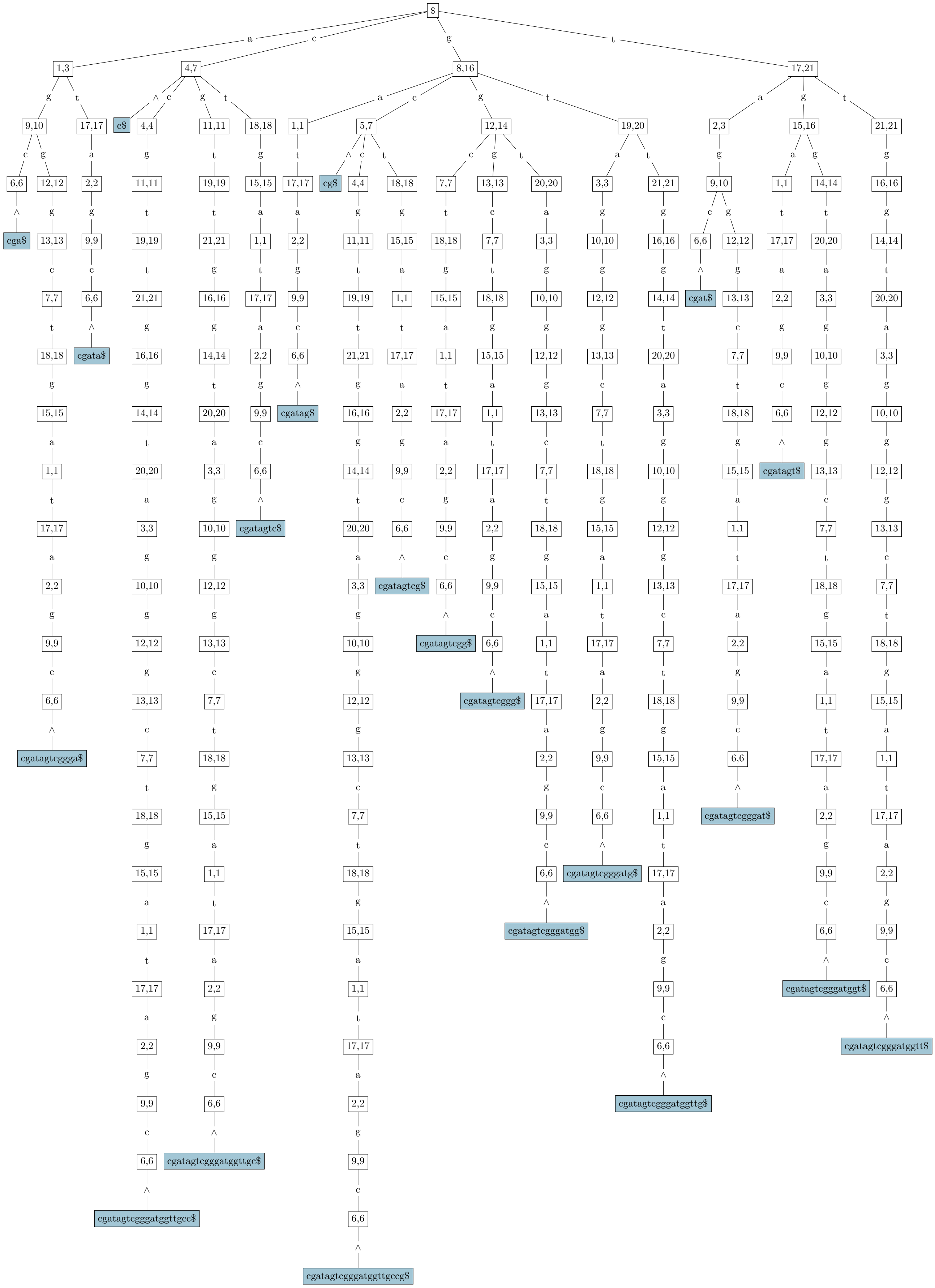

Figura 3: Árbol de prefijos con intervalos SA de la palabra “cgatagtcgggatggttgccg$”.

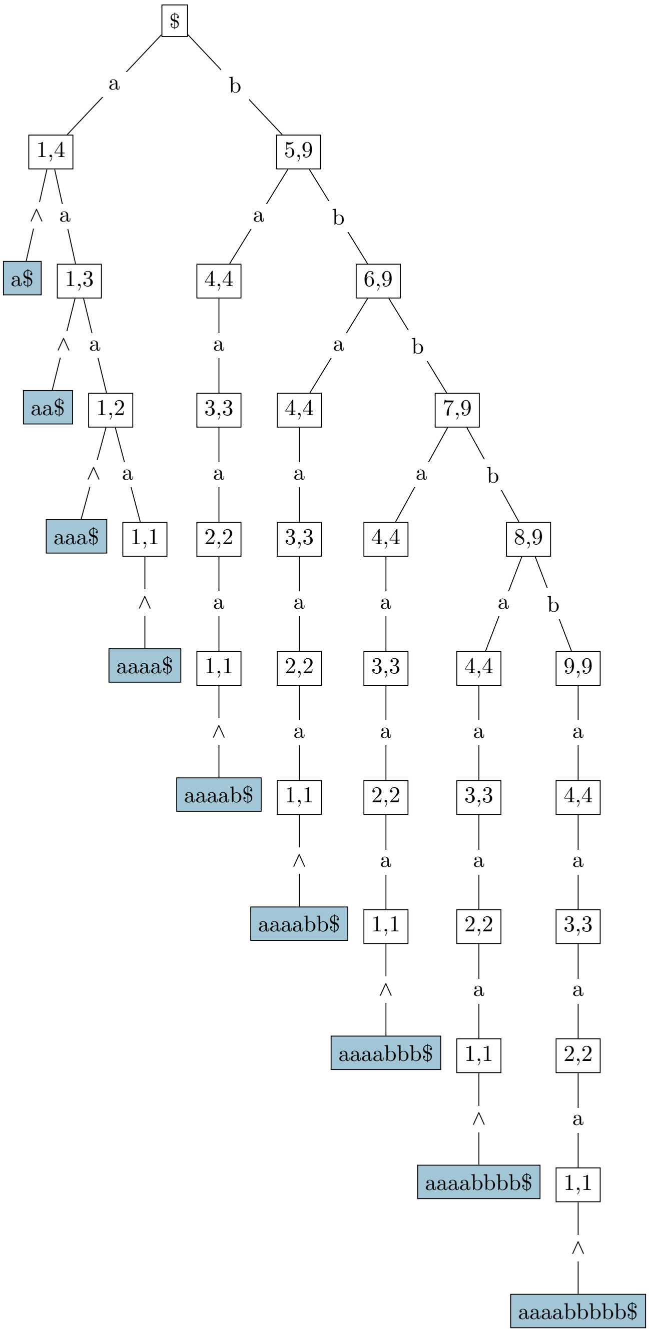

Figura 4: Árbol de prefijos con intervalos SA de la palabra "aaaabbbbb$".