Computabilidad y Complejidad:

Computación por membranas

Por: Ramon Ruiz Dolz
Salvador Martí Román

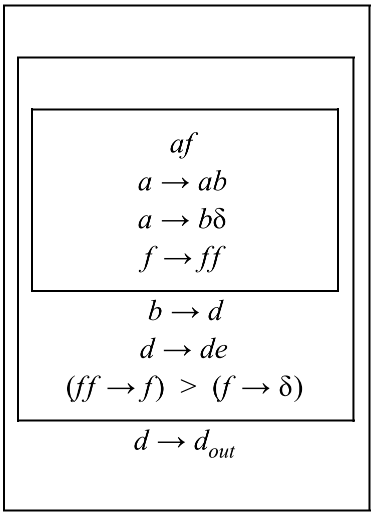
Laboratorio CMC

Actividad 1

En esta primera parte de la práctica se nos pide que especifiquemos el conjunto de los naturales que calculan los sistemas “a” y “b”. Para ello hemos realizado una traza del sistema de membranas en forma de árbol de posibilidades teniendo en cuenta sus diferentes resultados. Estas trazas a continuación no están completas ya que para ambos problemas hay un número infinito de naturales en el conjunto, no obstante hemos retratado un set de resultados que nos permitirá responder la pregunta.

1. Especifique el conjunto de naturales calculado por cada uno de los siguientes sistemas P

A)



El sistema “a” está compuesto por 3 membranas cada una con un set diferente de reglas. Estas transiciones no solo cambian la cadena pero también modifican la estructura del sistema “$a \rightarrow b\delta$” por ejemplo no solo transforma el símbolo “a” en “b” pero también hace que la membrana 3 (La más interior) desaparezca haciendo así que la membrana 2 herede la cadena de la membrana 1. Otro detalle a tener en cuenta son las prioridades en la ejecución aunque prácticamente todas tienen la misma prioridad podemos ver en la transición $(ff \rightarrow f) > (f \rightarrow \delta)$ que $(ff \rightarrow f)$ tiene mayor prioridad, es decir, no se realizará $(f \rightarrow \delta)$ si se puede realizar la operación de mayor prioridad.

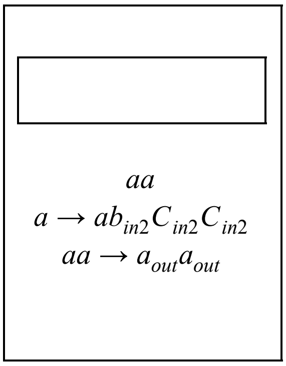
Adicionalmente, el problema indica que la membrana de salida en la cual se halla el resultado es la membrana exterior $i_0 = 1$. Tras terminar la ejecución podemos ver que los números generados pertenecen al conjunto de los números naturales $y \mid y \in x^2$ es decir el resultado pertenece a los números cuadrados. $\{1.4.9.16...\}$. Estos números vienen dados por la longitud de la cadena resultante en la membrana de salida.

La traza para la ejecución del sistema “a” se encuentra a continuación:
(documento adjunto a parte por cuestiones de formato)

Memoria 3: Computación por membranas

Laboratorio CMC

B)

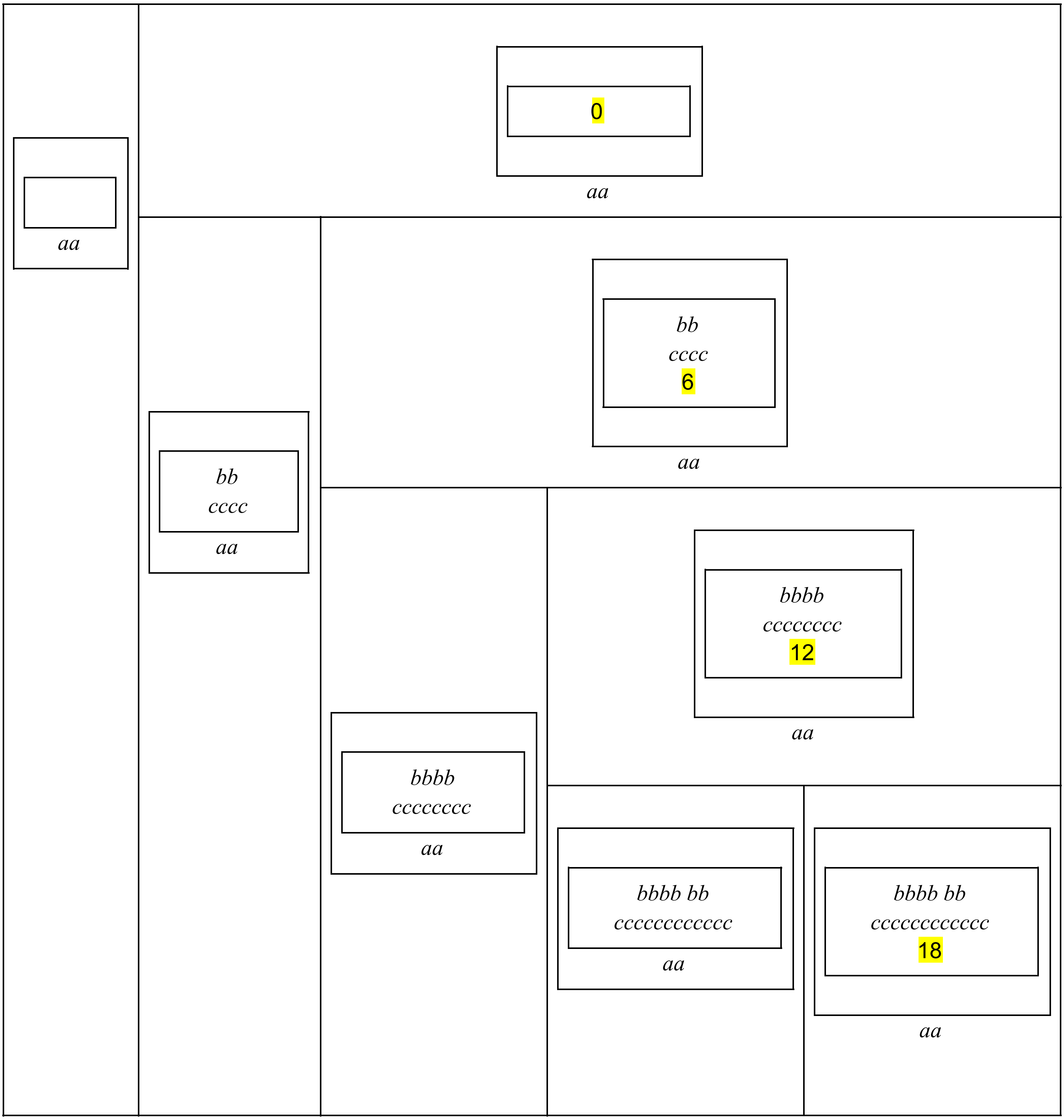


Este sistema “b” aunque más simple tiene un transformaciones no descritas en “a” que son las que transfieren caracteres entre membranas. Las que tienen la notación X_{in2} introducen las “proteínas” dentro de la membrana 2, este proceso en biología es llamado Endocitosis. Las transiciones con la notación X_{out} tienen la función contraria, la Exocitosis, expulsan de una región las “proteínas”. Esta vez la membrana de salida es la interior.

Este sistema produce el conjunto de los números naturales múltiplos de 6. El set reducido de respuestas que hemos calculado es: $0, 6, 12.18$ La traza realizada se encuentra en forma de tabla en la siguiente página.

Laboratorio CMC

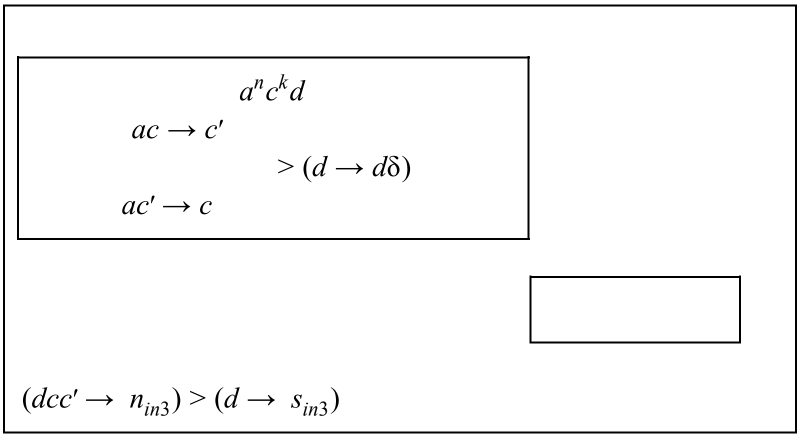
Traza del sistema de membranas “b”



Actividad 2

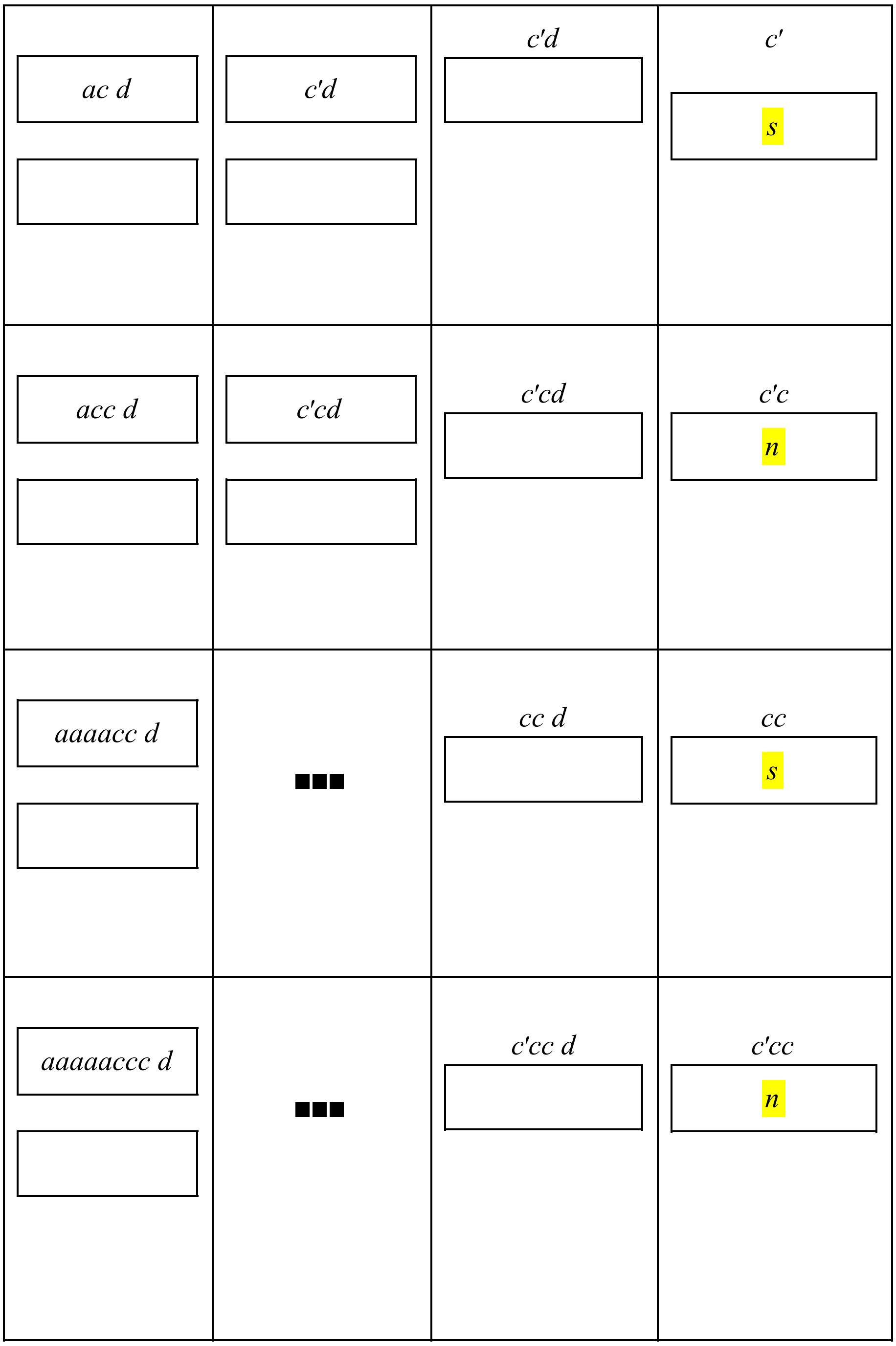
Dado el siguiente sistema P, establezca cuándo el sistema calcula como salida “s” y cuándo calcula como salida “n” (considere la región número 3 como la de salida)

En esta ocasión tenemos un sistema que no produce múltiples posibilidades pero dónde podemos cambiar la cadena de entrada modificando el la cantidad de “a” y “c” representado por “n” y “k” respectivamente en la configuración inicial.



El resultado de esta ejecución el cual se halla en la membrana interior de la derecha (3) solo pueden ser las cadenas s o n. El criterio que sigue este sistema es el de la división entera de el número n por el número k. Si la división entera resultante no deja ningún resto entonces la cadena s estará en la membrana de salida. Si lo contrario es cierto, es decir la división no es entera y resulta en un resto, la cadena n estará en la membrana de salida.

Para demostrar esto a continuación incluimos una traza de un grupo reducido pero ilustrativo de combinaciones de combinaciones de los valores “n” y “k”.


ac d
c'd
c'd
c'
s
acc d
c'cd
c'cd
c'c
n
aaaacc d
cc d
cc
s
aaaaaccc d
c'cc d
c'cc
n

Actividad 3

En esta última parte de la práctica se nos pide diseñar un módulo de matemática que recibiendo como único parámetro un número “n” devuelve la configuración del sistema de membranas “b” en el paso “n”. Esto se ha realizado de la siguiente manera:

Primero se hacen las asignaciones propias de la configuración inicial de “b”: el alfabeto de objetos “O”, la estructura de membranas “mu”, las cadenas asociadas a las membranas (w1 y w2), las reglas de transición (regla1 y regla2) y las membranas de salida (salida).

Después de la asignación de valores inicial se entra dentro de un bucle for delimitado con el parámetro step que determinará cuántas iteraciones del sistema de membranas se hacen. En cada iteración del bucle se calcula un número aleatorio entre 0 y 1 el cual se redondea después para simular el entorno no-determinista de las membranas. De tal manera si el número de decisión es igual a 0 se aplica la primera regla de la membrana 1.

Si el número de decisión fuera 1 primero se trataría de aplicar la segunda regla la cual hace matching con dos “a” y si no se pudiera aplicar se intenta aplicar la primera regla. Aunque estemos usando Strings para la representación de las membranas este algoritmo no hace matching con “aa” debido a que el orden de los caracteres no se debe tener en cuenta en computación con membranas.

Una vez hechos los reemplazos nos falta mover las “proteínas” a sus membranas correspondientes ya que nuestra membrana w1 contiene caracteres como “`Bin2`” ó ”`Cin2`". Para ello hacemos un contado de esos términos, los borramos y copiamos con la forma `"B"` y `"C"` dentro de la membrana w2.

Estos pasos son repetidos hasta que se han hecho todos los pasos deseados y finalmente se imprime la configuración del sistema de membranas. Este estado no es necesariamente el final. A continuación se incluyen las 3 configuraciones posibles del sistema con paso 2.

```
celularSeis[2]

{{A,B,C},{{}},,BBCCCC,{{A,ABin2Cin2Cin2},{AA,AoutAout}},{{}},2}

{{A,B,C},{{}},,,{{A,ABin2Cin2Cin2},{AA,AoutAout}},{{}},2}

{{A,B,C},{{}},AA,BBCCCCBBCCCC,{{A,ABin2Cin2Cin2},{AA,AoutAout}},{{}},2}
```

Código:

```
celularSeis[step_] :=
  Module[{O, mu, w1, w2, regla1, regla2, salida, n, decision, match,
    count, amount, transport, transported, transportCount, i, j},
   O = {"A", "B", "C"};
   mu = {{}};
   w1 = "AA";
   w2 = "";
   regla1 = {{"A", "ABin2Cin2Cin2"}, {"AA", "AoutAout"}};
   regla2 = {{}};
   salida = 2;
   transport = {"Bin2", "Cin2"};
   transported = {"B", "C"};

   (*Este es el For que determina la cantidad de pasos que evaluaremos*)
   For[n = 1, n <= step, n++,
    decision = Round[RandomReal[]];
    amount = StringCount[w1, "A"];
    If[decision == 0,
     (*True*)
     If[amount >= 1,
       (*Intentar primera transición primero*)
       w1 = StringReplace[w1, regla1[[1]][[1]] -> regla1[[1]][[2]]];];,
     (*False*)
     If[amount >= 2,
       (*Intentar segunda transición primero*)
       w1 = StringReplace[w1, "A" -> "", Floor[amount, 2]];
       (*Esta línea borra la mayor cantidad de A's posible siempre que ese numero sea
par
       Es equivalente a buscar AA porque el orden de los caracteres no importa.*)
       ,
       (* Si no se puede la segunda hace la primera (Ningún efecto si no hay a)*)
       w1 =  StringReplace[w1, regla1[[1]][[1]] -> regla1[[1]][[2]]];
       ];
     ];

    For[j = 1, j <= Length[transport], j++,
     transportCount = StringCount[w1, transport[[j]]];
     w1 = StringDelete[w1, transport[[j]]];
     For[i = 1, i <= transportCount, i++,
      w2 = StringJoin[w2, transported[[j]]];
      ];
     ];
    ];
   Print[{O, mu, w1, w2, regla1, regla2, salida}];
   ];
```