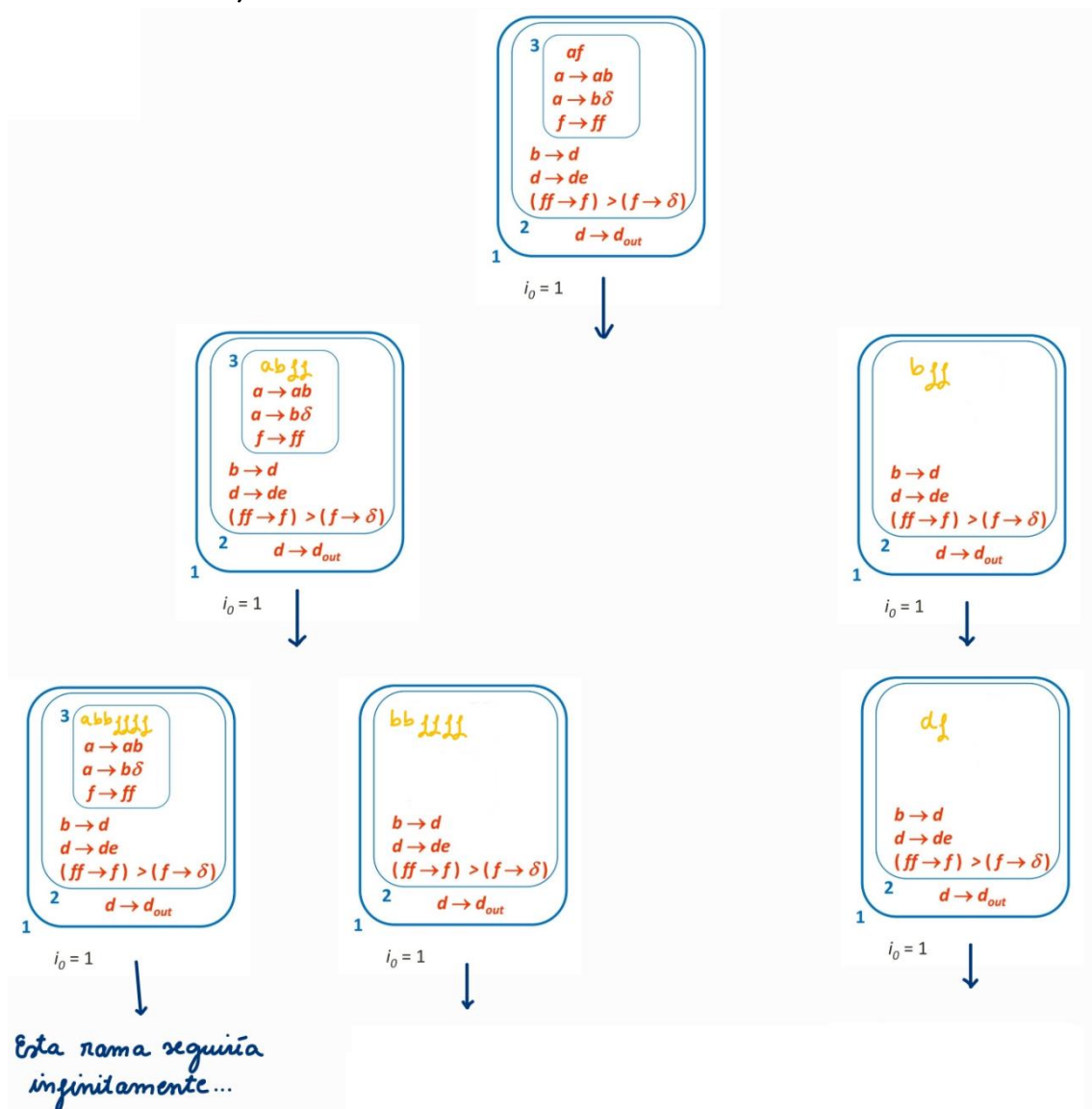


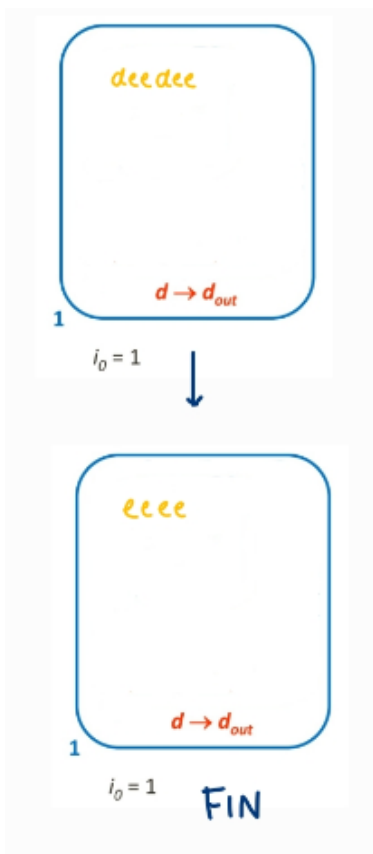
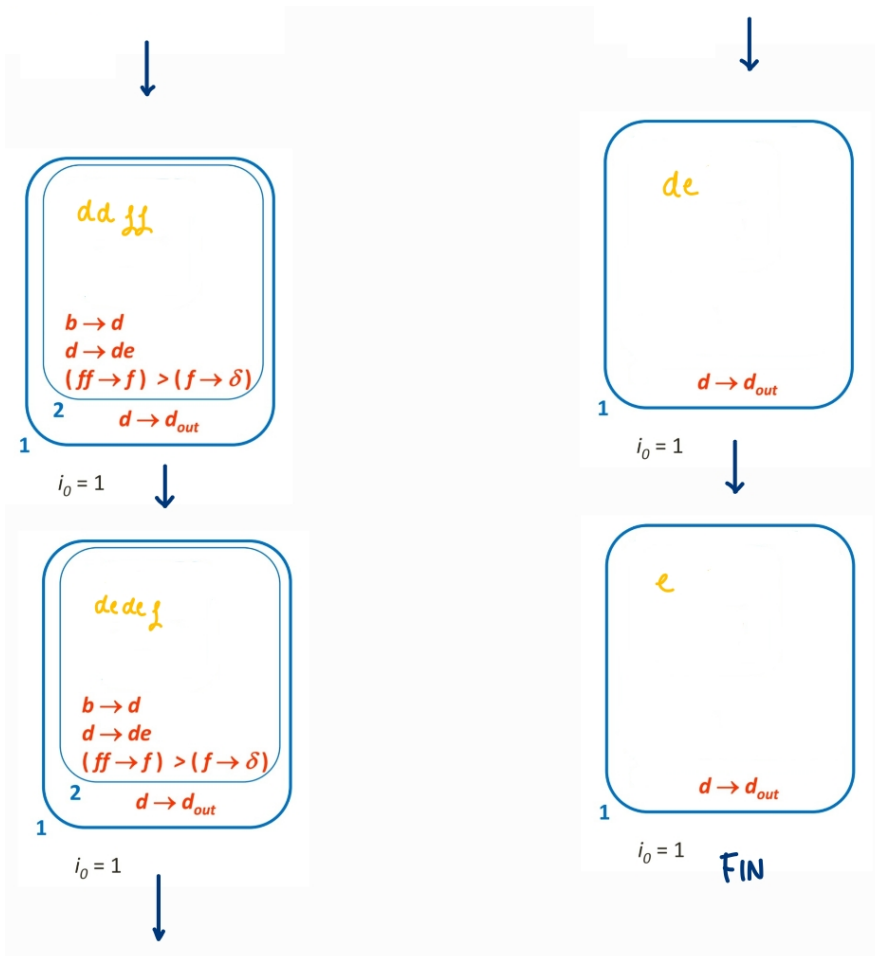
# Práctica 3 - Computación con membranas

Iñaki Diez Lambies y Aitana Menárguez Box

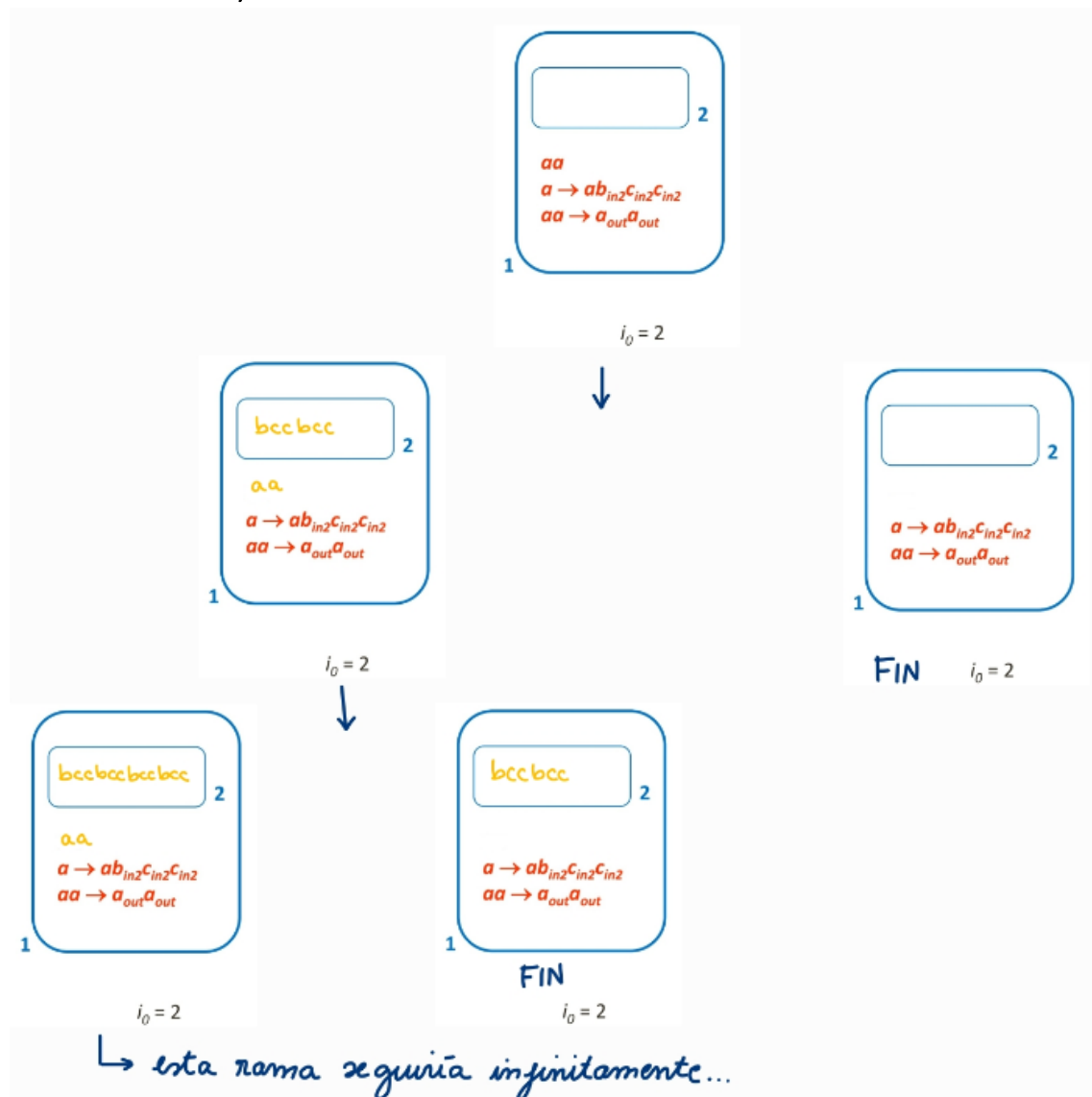
## 1 EJERCICIO 1

### 1.1 APARTADO A)



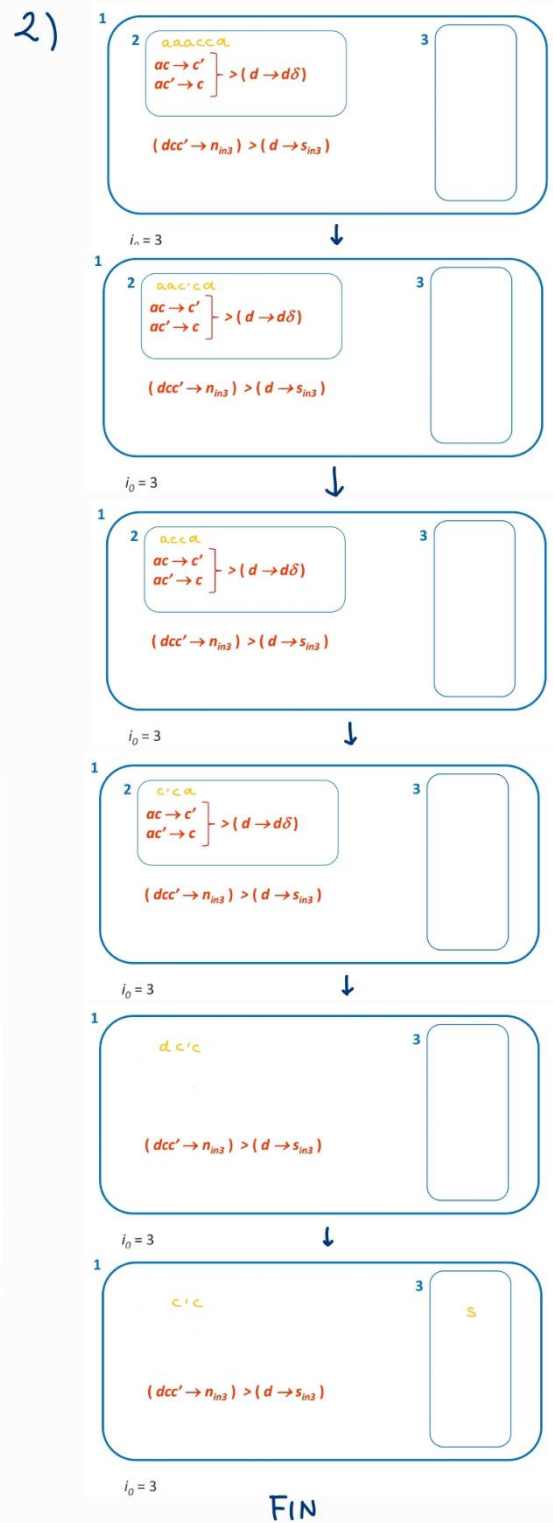
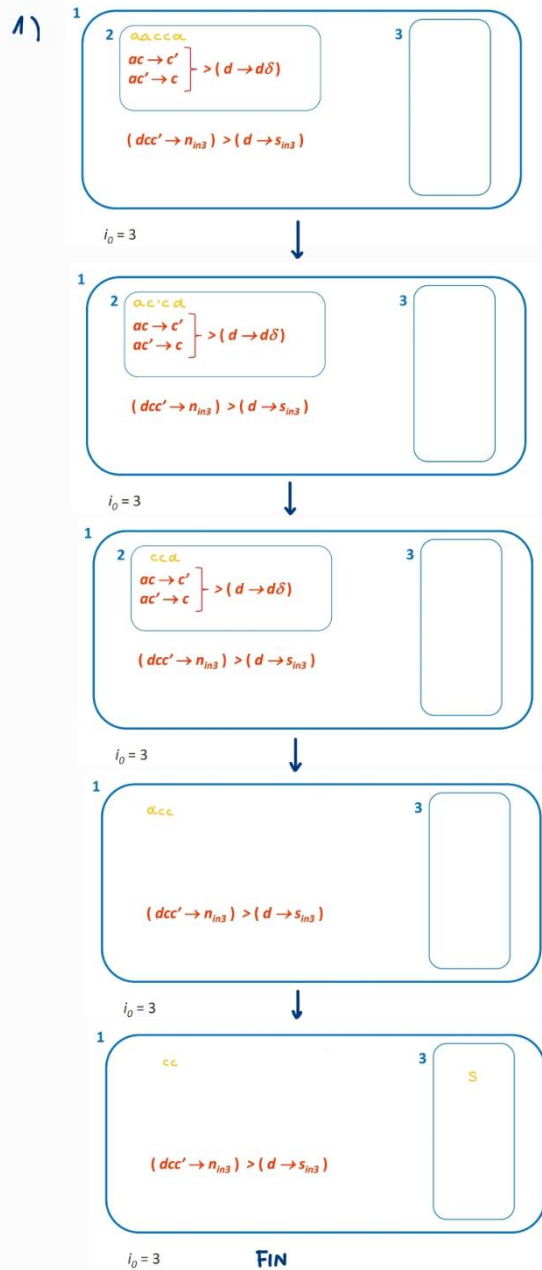


## 1.2 APARTADO B)

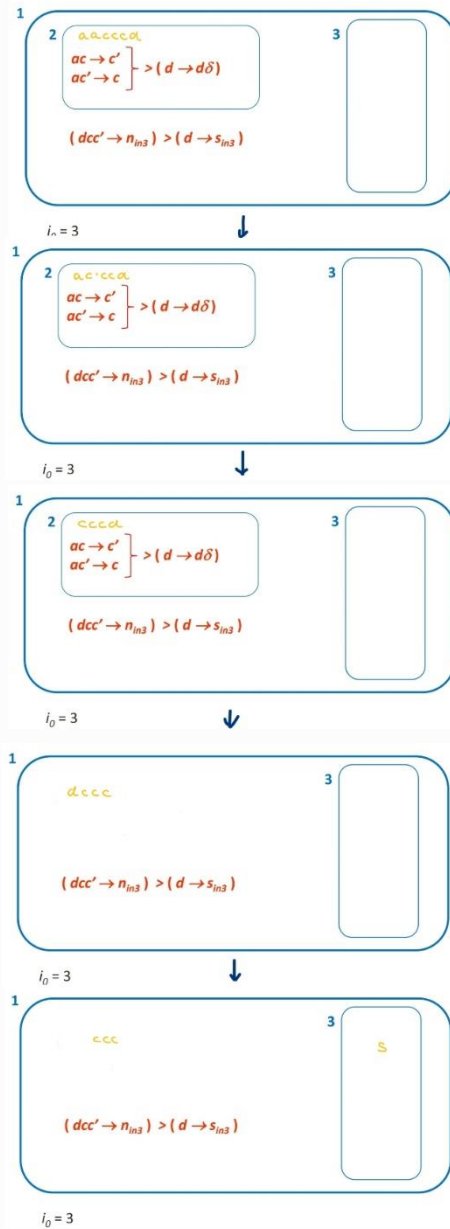


## 2 EJERCICIO 2

Aquí proponemos diferentes posibilidades para ver el comportamiento del sistema P.



3)



FIN

### 3 EJERCICIO 3

---

En este ejercicio se nos pedía diseñar un módulo de Mathematica que, dado un valor entero  $n$ , nos proporcione la configuración del sistema P descrito.

Para realizar esto hemos analizado la naturaleza y comportamiento del sistema en cuestión y nos hemos percatado que el sistema trata de generar cadenas  $[b, c, c, b, c, c]$  con un 50% de probabilidad de escoger entre hacer esto o extraer fuera de su membrana más exterior su cadena "aa" y, por tanto, dejar la membrana 1 sin contenido. Esto acabaría por dejar al sistema P sin nada que hacer.

A continuación, tenemos el código comentado:

```
Ej3[n_] := Module[{i, choice, res},  
  módulo  
  res = {a, a, {}}; (* Estado inicial del sistema P *)  
  For[i = 0, i < n, i++, (* Realizamos N transiciones *)  
    para cada  
    choice = Random[Integer, 1]; (* Elección entre la primera y la segunda regla *)  
    aleatorio entero  
    If[choice == 0, (* Elección de la regla con entrada 'a' *)  
      si  
      AppendTo[res[[3]], {b, c, c, b, c, c}]; (* Se ejecuta 2 veces *)  
      añade al final  
  
      res[[3]] = Flatten[res[[3]]];  
      aplana  
    ];  
    If[choice == 1, (* Elección de la regla con entrada 'aa' *)  
      si  
      res = {res[[3]]};  
      Break[]; (* Si vaciamos no hay reglas aplicables en ningún caso *)  
      finaliza iteración  
    ];  
  
  ];  
  Return[res];  
  retorna  
];
```