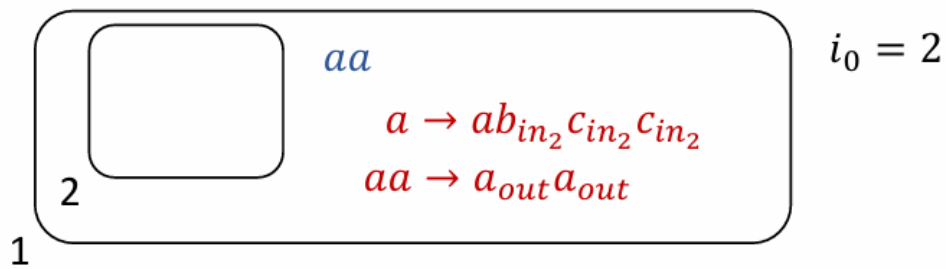


Ejercicio 1 --- Simulación del sistema P dado tras “n” transiciones:



```

In[46]:= RunP1b[numSteps_Integer?NonNegative] := Module[
    {
        (*Contador de transiciones*)
        step = 0,
        (*Multiconjunto de la membrana 2*)
        reg2 = {},
        (*Variables auxiliares*)
        exp, resultString
    },
    (**1) Aplicación de las reglas durante numSteps pasos**)
    For[step = 0, step < numSteps, step++,
        (*Regla1:con probabilidad ½ añadimos bccbcc y detenemos*)
        If[RandomInteger[{0, 1}] == 0,
            reg2 = Join[reg2, {"b", "c", "c", "b", "c", "c"}];
            Break[];
        ];
        (*Regla2:en caso contrario seguimos al siguiente paso*)
    ];
    (**2) Construcción del string de salida**)
    exp = 2 * step;
    resultString = If[
        step == 0,
        "", (*ninguna regla aplicada*)
        StringRepeat["bcc", exp] (*"bcc"^exp*)
    ];
    (**3) Mostrar resultados**)
    Print["\nNúmero de transiciones aplicadas: ", step,
        "\nRegion2: ",
        If[resultString == "", "{}", resultString], "\n"
    ];
    resultString
];

```

```

In[18]:= RunP1b[1]

```

```

Número de transiciones aplicadas: 1
Region2: bccbcc

```

```

Out[18]= bccbcc

```

```
In[19]:= RunP1b[1]
```

Número de transiciones aplicadas: 0
Region2: {}

```
Out[19]=
```

```
In[25]:= RunP1b[50]
```

Número de transiciones aplicadas: 3
Region2: bccbccbccbccbcc

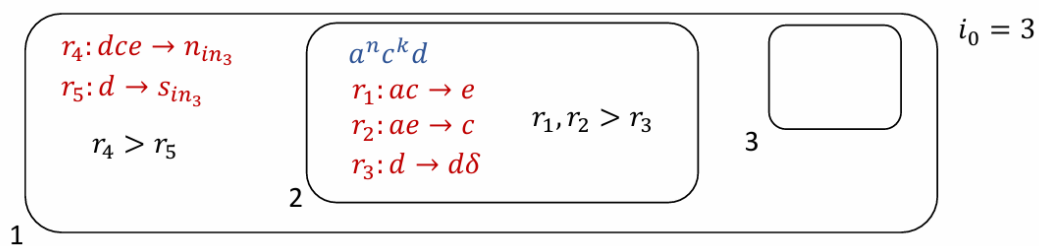
```
Out[25]= bccbccbccbccbcc
```

```
In[38]:= RunP1b[50]
```

Número de transiciones aplicadas: 7
Region2: bccbccbccbccbccbccbccbccbccbccbccbcc

```
Out[38]= bccbccbccbccbccbccbccbccbccbccbccbccbcc
```

Ejercicio 2 --- Simulación del sistema P dado:



```
In[39]:= RunDivisibilityP[n_Integer?NonNegative, k_Integer?NonNegative] :=
```

```
Module[
  |módulo
```

```
{a2 = n, c2 = k, d2 = 1, e2 = 0,
  a1 = 0, c1 = 0, d1 = 0, e1 = 0,
  n3 = 0, s3 = 0,
  t1, t2, t},
```

```
(*1) Membrana 2*)
```

```
While[True,
```

```
  |mient... |verdadero
```

```
  t1 = Min[a2, c2];
  |mínimo
```

```

    t2 = Min[a2 - t1, e2];
    If[t1 + t2 > 0,
    a2 -= (t1 + t2);
    c2 -= t1; e2 += t1;
    e2 -= t2; c2 += t2;
    Continue[];
    ];
    If[d2 > 0,
    Break[]];
    Break[];
    ];
    (*2) Disolución de 2*)
    a1 += a2; c1 += c2; d1 += d2; e1 += e2;
    a2 = c2 = d2 = e2 = 0;
    (*3) Membrana 1*)
    t = Min[d1, c1, e1];
    If[t > 0,
    d1 -= t; c1 -= t; e1 -= t;
    n3 += t];
    If[d1 > 0,
    s3 += d1;
    d1 = 0];
    (*4) Construcción de la salida final*)
    With[
    {
    objs1 = Join[
    ConstantArray["c", c1],
    ConstantArray["e", e1],
    ],
    objs3 = Join[
    ConstantArray["n", n3],
    ConstantArray["s", s3],
    ],
    },
    {Region1, Sequence @@ objs1, {Region3, Sequence @@ objs3}}

```

```

]
];

```

```
In[40]:= RunDivisibilityP[17, 5]
```

```
Out[40]= {Region1, c, e, e, {Region3, n}}
```

```
In[41]:= RunDivisibilityP[20, 5]
```

```
Out[41]= {Region1, c, c, c, c, c, {Region3, s}}
```

```
In[42]:= RunDivisibilityP[5, 5]
```

```
Out[42]= {Region1, e, e, e, e, e, {Region3, s}}
```

```
In[43]:= RunDivisibilityP[4, 5]
```

```
Out[43]= {Region1, e, e, e, {Region3, n}}
```

```
In[44]:= RunDivisibilityP[500, 5]
```

```
Out[44]= {Region1, c, c, c, c, c, {Region3, s}}
```

```
In[45]:= RunDivisibilityP[16, 5]
```

```
Out[45]= {Region1, e, e, e, {Region3, n}}
```