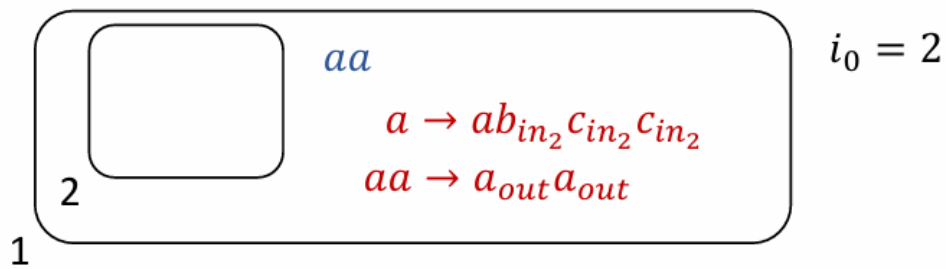


Ejercicio 1 --- Simulación del sistema P dado tras “n” transiciones:



```

In[64]:= 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 1/2 añadimos bccbcc y detenemos*)
        If[RandomInteger[{0, 1}] == 0,
            reg2 = Join[reg2, {"b", "c", "c", "b", "c", "c"}];
            Break[];
        ];
    ];
    (**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,
        "\nMulticonjunto en membrana 2: ",
        If[resultString == "", "{}", resultString], "\n"
    ];
    resultString
];

```

```

In[89]:= RunP1b[1]

```

```

Número de transiciones aplicadas: 1
Multiconjunto en membrana 2: bccbcc

```

```

Out[89]= bccbcc

```

In[90]:= RunP1b[1]

Número de transiciones aplicadas: 0
Multiconjunto en membrana 2: {}

In[91]:= RunP1b[50]

Número de transiciones aplicadas: 1
Multiconjunto en membrana 2: bccbcc

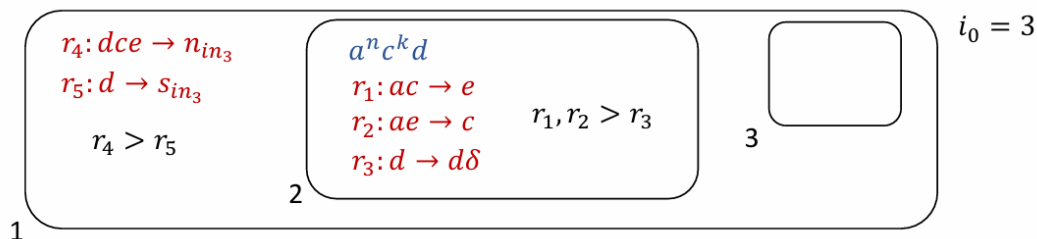
Out[91]= bccbcc

In[112]:= RunP1b[50]

Número de transiciones aplicadas: 8
Multiconjunto en membrana 2: bccbcbccbccbccbccbccbccbccbccbccbccbccbccbcc

Out[112]= bccbcbccbccbccbccbccbccbccbccbccbccbccbccbccbccbccbcc

Ejercicio 2 --- Simulación del sistema P dado:



In[45]:= RunDivisibilityP[n_Integer?NonNegative, k_Integer?NonNegative] :=

```

Module[
  [módulo]
  {(*Variables internas*)
    [variables]
    (*Membrana 2: a^n c^k d*)
    a2 = n, c2 = k, d2 = 1, e2 = 0,
    (*Membrana 1: inicialmente vacía*)
    a1 = 0, c1 = 0, d1 = 0, e1 = 0,
    (*Membrana 3 (salida): contendrá "n" o "s"*)
    n3 = 0, s3 = 0, t1, t2, t},

  (*1) Simulación en membrana 2*)

```

```

While[True,
  mientras Verdadero
  (*Modo maximally-parallel: r1 y r2 primero*)
  t1 = Min[a2, c2]; (*ac→e*)
  mínimo
  t2 = Min[a2 - t1, e2]; (*ae→c*)
  mínimo
  If[t1 + t2 > 0,
    si
    (*Aplico r1 t1 veces y r2 t2 veces*)
    a2 -= (t1 + t2);
    c2 -= t1; e2 += t1;
    e2 -= t2; c2 += t2;
    Continue[];
    continúa iteración
  ];
  (*Si no caben r1 ni r2, pruebo r3*)
  If[d2 > 0,
    si
    (*r3: d→d+δ (disuelve)*)
    Break[]; (*marcamos fin de la 2*)
    finaliza iteración
  ];
  Break[];
  finaliza iteración
];

(*2) Disolución de la membrana 2*)
a1 += a2; c1 += c2; d1 += d2; e1 += e2;
a2 = c2 = d2 = e2 = 0;

(*3) Simulación en membrana 1*)
(*r4: d c e→n_in3 (prioridad sobre r5)*)
t = Min[d1, c1, e1];
mínimo
If[t > 0,
  si
  d1 -= t; c1 -= t; e1 -= t;
  n3 += t;
];
(*r5: d→s_in3 (si ya no cabe r4)*)
If[d1 > 0,
  si
  s3 += d1;
  d1 = 0;
];

(*4) Construir la salida*) {
  (*Región 1*)
  {"Region1", Flatten@{
    aplana
    Table["a", {a1}],
    tabla
  }
}

```

```

      Table["c", {c1}],
      Table["d", {d1}],
      Table["e", {e1}]
    },
    (*Región 2 (disuelta)*)
    {"Region2", {}},
    (*Región 3*)
    {"Region3", Flatten@
      Table["n", {n3}],
      Table["s", {s3}]
    }
  }
];

```

In[46]:= RunDivisibilityP[17, 5]

Out[46]= {{Region1, {c, e, e}}, {Region2, {}}, {Region3, {n}}}

In[47]:= RunDivisibilityP[20, 5]

Out[47]= {{Region1, {c, c, c, c, c}}, {Region2, {}}, {Region3, {s}}}

In[48]:= RunDivisibilityP[5, 5]

Out[48]= {{Region1, {e, e, e, e, e}}, {Region2, {}}, {Region3, {s}}}

In[49]:= RunDivisibilityP[4, 5]

Out[49]= {{Region1, {e, e, e}}, {Region2, {}}, {Region3, {n}}}

In[50]:= RunDivisibilityP[500, 5]

Out[50]= {{Region1, {c, c, c, c, c}}, {Region2, {}}, {Region3, {s}}}

In[51]:= RunDivisibilityP[16, 5]

Out[51]= {{Region1, {e, e, e}}, {Region2, {}}, {Region3, {n}}}