Computabilidad y Complejidad (CMC)

Práctica 4: Computación con membranas. Sistemas P

José M. Sempere

Valencian Research Institute for Artificial Intelligence (VRAIN)
Valencian Graduate School and Research Network of Artificial Intelligence (valgrAI)
Department of Informatics Systems and Computation (DSIC)
Universitat Politècnica de València (UPV)

http:// jsempere.webs.upv.es

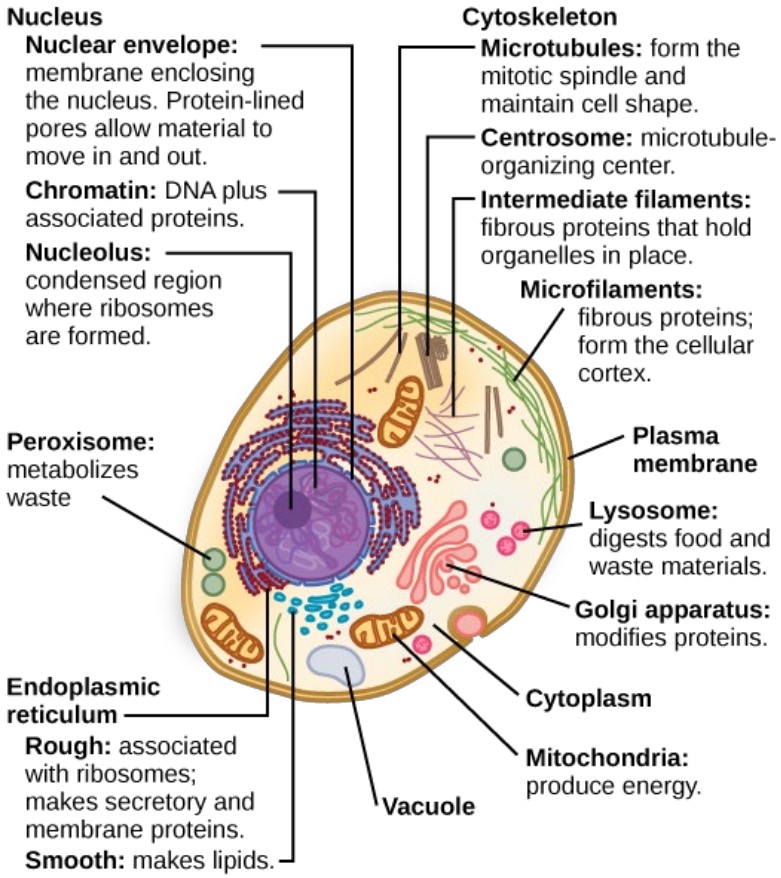
jsempere@dsic.upv.es

Índice:

Bibliografía Básica

- Conceptos de genética. W. Clug, M. Cummings. Prentice Hall (5a. ed.). 1999.
- Membrane computing. An introduction. Gh. Păun. Springer. 2002.
- Computing with Bio-Molecules. Gh. Păun, (editor). Springer. 1998.
- The P systems web page (http://ppage.psystems.eu/)

Estructura de la célula animal eucariota



https://bio.libretexts.org

La célula viva puede considerarse un mecanismo real de procesamiento de información codificada bioquímicamente.

Multiconjuntos y proyecciones

Un *multiconjunto* es un conjunto con multiciplicidad asociada a sus elementos.

Dado el conjunto *V* un multiconjunto *M* sobre *V* se define mediante $M: V \longrightarrow \mathbb{N}$

- $M(a)$ es la *multiciplicidad* del elemento a
- $supp(M) = \{ a \in V : M(a) \geq 0\}$ es el *soporte* de M.
- $M_1 \subseteq M_2$ si $\forall\ a \in V\ M_1(a) \leq M_2(a)$
- $(M_1 \cup M_2)(a) = M_1(a) + M_2(a)$
- $(M_1 - M_2)(a) = M_1(a) - M_2(a)$ (sólo se define si $M_2 \subseteq M_1$)
- La *amplificación* de M por n (≥ 0) es $(n \otimes M)(a) = n\ M(a)\ \forall\ a \in V$
- Dado el multiconjunto M y dados $Y \subseteq X$ se define la *proyección* de M sobre Y como

$$pr_Y M(a) = \begin{cases} M(a) \text{ si } a \in Y \\ 0 \text{ en otro caso} \end{cases}$$

Todo multiconjunto con soporte finito admite una representación mediante la cadena $a_1{}^{M(a_1)} a_2{}^{M(a_2)} \ldots a_n{}^{M(a_n)}$ (o cualquiera de sus permutaciones)

Multiconjuntos y proyecciones

Dado un alfabeto $\Sigma = \{a_1, a_2, \dots, a_n\}$ y una cadena $x \in \Sigma^*$ se define la proyección de Parikh de *x* en Σ como la aplicación $\Psi_\Sigma: \Sigma^* \longrightarrow \mathbb{N}^n$

$$\Psi_\Sigma(x) = (|x|_{a_1}, |x|_{a_2}, \dots, |x|_{a_n})$$

Dado un lenguaje $L \subseteq \Sigma^*$ $\Psi_\Sigma(L) = \{ \Psi_\Sigma(x) : x \in L \}$

Dada una familia de lenguajes FL *Ps*FL es la familia de conjuntos de proyecciones de Parikh de los lenguajes L pertenecientes a FL

Conjuntos naturales

Dado un alfabeto $\Sigma = \{a_1, a_2, \dots, a_n\}$ y un lenguaje $L \subseteq \Sigma^*$ se define el conjunto de naturales de L como

$$NL = \{ n \in N : x \in L \wedge |x| = n \}$$

Dada una familia de lenguajes FL, NFL es el conjunto de los conjuntos de naturales definidos por lenguajes de FL.

$$NFIN \subset NREG = NLIN = NCF \subset NCS \subset NRE$$

Computación con membranas. Sistemas P

Los sistemas P o modelos de computación con membranas fueron introducidos por Gh. Paun en 1998. Se inspiran en el funcionamiento de la célula eucariota animal y es un modelo de computación paralelo, distribuido y no determinista. Los principales ingredientes que se incorporan en el modelo son:

- Compartimentación en regiones separadas por membranas (cada región es un espacio de trabajo en el que operan reglas predefinidas)
- La información se representa mediante símbolos o cadenas, así como estructuras más complejas.
- La computación se basa en sistemas de reescritura (se asimila a las reacciones que suceden en la célula a nivel bioquímico)
- Existe la posibilidad de comunicar la información entre las regiones y de crear y destruir regiones dentro del modelo.

Los sistemas P se pueden considerar modelos de computación no convencionales. Forman parte de la computación bio-inspirada y son un paradigma de computación celular. Otros modelos celulares son las redes neuronales y las redes de procesadores bio-inspirados (además de los modelos clásicos de autómatas celulares).

Algunos tipos de sistemas P

- Sistemas P de transición
- Sistemas P en tejidos
- Sistemas P catalíticos
- Sistemas P con comunicación
- Autómatas P
- Sistemas P con cadenas
- Sistemas P en splicing
- Sistemas P numéricos (*conformon P systems*)
- Sistemas P metabólicos
- Sistemas P con membranas activas y objetos en las membranas
- Sistemas P con "impulsos neuronales" (*spiking neuron P systems*)
- Sistemas P estocásticos y probabilistas
- Sistemas P en colonias
- etc, etc.

En esta práctica nos centraremos sólo en los sistemas P de transición que fueron los primeros propuestos por Paun. Además, permiten la universalidad con estructuras de membranas muy simples.

Sistemas P de transición

Sistema P de transición de grado m

$$\Pi = (O, \mu, w_1, w_2, \dots, w_m, (R_1, \rho_1), (R_2, \rho_2), \dots, (R_m, \rho_m), i_o)$$

- O alfabeto de objetos
- μ estructura de membranas (representable mediante un árbol)
- $w_1, \dots, w_m$ cadenas asociadas a cada región delimitada por cada membrana (multiconjuntos iniciales)
- $(R_1, \rho_1), \dots, (R_m, \rho_m)$ reglas y relaciones de prioridad (orden parcial entre las reglas)
- $i_0 \in \{1, \dots, m\} \cup \{\infty\}$ membrana de salida

Reglas

$C = \{a_{here}, a_{out}, a_{in_j} : a \in O, j \in \{1, .. m\}\}$ (omitiremos el direccionamiento $here$) (la etiqueta in_j denota direccionamiento por objetivo)

- $u \longrightarrow v$ (reglas de evolución sin disolución)
- $u \longrightarrow v\delta$ (reglas de evolución con disolución)

$u \in O^*$ $v \in C^*$ $\delta \notin O$ (disolución de membranas)

Dada la regla $u \to v$ ó $u \to v\delta$ llamaremos radio de la regla a $|u|$ (longitud de u)

Un sistema Π es *cooperativo* si contiene alguna regla con radio mayor que 1

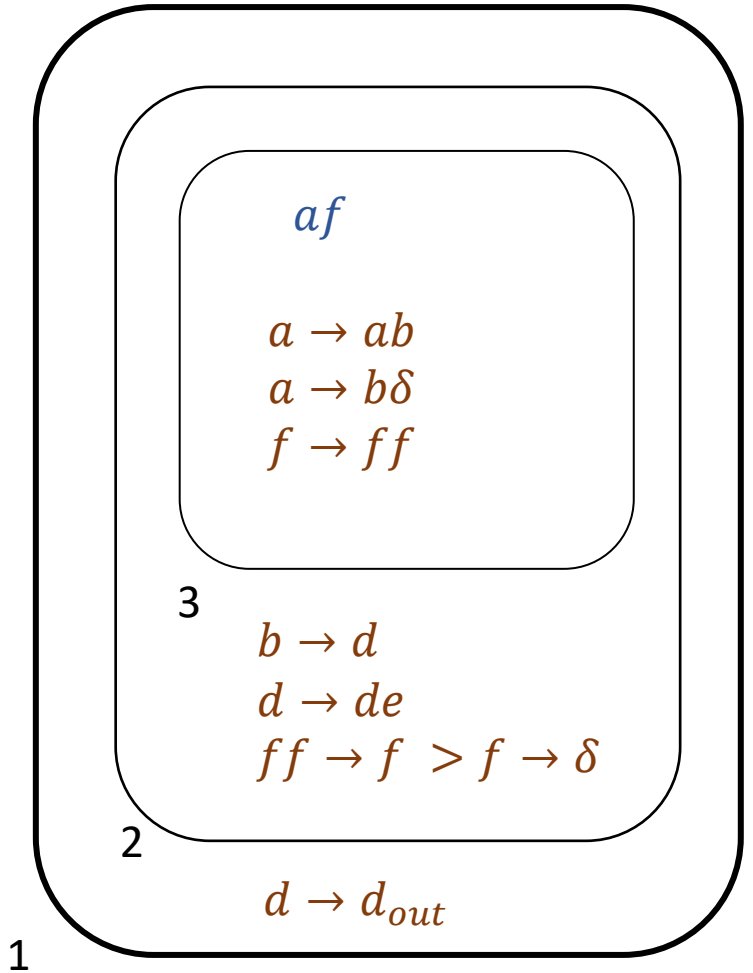
Sistemas P de transición (representación mediante diagramas)

$$\Pi = (O, \mu, w_1, w_2, \ldots, w_m, (R_1, \rho_1), (R_2, \rho_2), \ldots, (R_m, \rho_m), i_o)$$

Ejemplo

Un sistema P de grado 3

$O = \{a, b, d, e, f\}$
$\mu = [_1 [_2 [_3]_3]_2]_1$
$w_1 = \lambda \quad R_1 = \{d \to d_{out}\} \quad \rho_1 = \emptyset$
$w_2 = \lambda \quad R_2 = \{b \to d, d \to de, r_1: ff \to f, r_2: f \to \delta\} \quad \rho_2 = \{r_1 > r_2\}$
$w_3 = af \quad R_3 = \{a \to ab, a \to b\delta, f \to ff\} \quad \rho_3 = \emptyset$
$i_0 = 1$



Configuraciones y transiciones

$$\Pi = (O, \mu, w_1, w_2, \ldots, w_m, (R_1, \rho_1), (R_2, \rho_2), \ldots, (R_m, \rho_m), i_o)$$

Configuraciones de los sistemas P

$(\mu', w_{i_1}, w_{i_2}, \ldots, w_{i_k})$

$(\mu, w_1, w_2, \ldots, w_m)$ configuración inicial de Π

Transiciones

$C_1 = (\mu', w_{i_1}, w_{i_2}, \ldots, w_{i_k})$

$C_2 = (\mu'', w_{j_1}, w_{j_2}, \ldots, w_{j_p})$

$C_1 \Rightarrow C_2$

El sistema para cuando no se puede aplicar ninguna regla en ninguna región

¿ como se aplican las reglas en los sistemas P ?

Configuraciones y transiciones

$$\Pi = (O, \mu, w_1, w_2, \ldots, w_m, (R_1, \rho_1), (R_2, \rho_2), \ldots, (R_m, \rho_m), i_o)$$

$$C_1 = (\mu', w_{i_1}, w_{i_2}, \ldots, w_{i_k})$$

$$C_2 = (\mu'', w_{j_1}, w_{j_2}, \ldots, w_{j_p})$$

$C_1 \Rightarrow C_2$

Una regla $u \rightarrow v$ es aplicable si los objetos que definen u se encuentran presentes en la región donde actúa la regla.

Modos de aplicación de las reglas (modos de derivación)

- Asíncrono : al menos se aplica una regla
- Secuencial : sólo se aplica una regla
- Máximamente paralelo : se aplica un multiconjunto no extensible de las reglas
- Máximamente paralelo sobre un número máximo de reglas : se aplica un multiconjunto no extensible de reglas con máxima cardinalidad en el número de reglas
- Máximamente paralelo sobre un número máximo de objetos : se aplica un multiconjunto no extensible de reglas que afecta al máximo número de objetos posible

Configuraciones y transiciones

$$\Pi = (O, \mu, w_1, w_2, \dots, w_m, (R_1, \rho_1), (R_2, \rho_2), \dots, (R_m, \rho_m), i_o)$$

$aaaccb$

$r_1: a \to d$

$r_2: a \to f$

$r_3: aa \to bc$

$r_4: accc \to dd$

$r_5: aaacc \to b$

Reglas aplicables: r_1, r_2, r_3, r_5

- Asíncrono : Se aplica alguno de los siguientes multiconjuntos de reglas
  $\{r_1, r_1, r_1\}, \{r_1\}, \{r_1, r_2\}, \{r_2, r_2, r_2\}, \{r_1, r_3\}, \dots$
- Secuencial : se aplica sólo una de las reglas r_1, r_2, r_3, r_5
- Máximamente paralelo : se aplican alguno de los siguientes multiconjuntos de reglas
  $\{r_1, r_1, r_1\}, \{r_1, r_1, r_2\}, \{r_1, r_2, r_2\}, \{r_2, r_2, r_2\}, \{r_1, r_3\}, \dots$
- Máximamente paralelo sobre un número máximo de reglas : se aplican alguno de los siguientes multiconjuntos de reglas
  $\{r_1, r_1, r_2\}, \{r_1, r_2, r_2\}, \{r_1, r_3\}$
- Máximamente paralelo sobre un número máximo de objetos : se aplica r_5

Configuraciones y transiciones

$$\Pi = (O, \mu, w_1, w_2, \dots, w_m, (R_1, \rho_1), (R_2, \rho_2), \dots, (R_m, \rho_m), i_o)$$

La prioridad de entre reglas aplicables bloquea la ejecución de las menos prioritarias

$aaaccb$

$r_1: a \to d$

$r_2: a \to f$

$r_3: aa \to bc$

$r_4: accc \to dd$

$r_5: aaacc \to b$

Reglas aplicables: r_1, r_2, r_3, r_5

prioridades: $r_3 > r_5 > r_1, r_2$

$aaaccb$

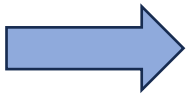
$r_1: a \to d$

$r_2: a \to f$

$r_3: aa \to bc$

$r_4: accc \to dd$

$r_5: aaacc \to b$



$bcaccb$

$r_1: a \to d$

$r_2: a \to f$

$r_3: aa \to bc$

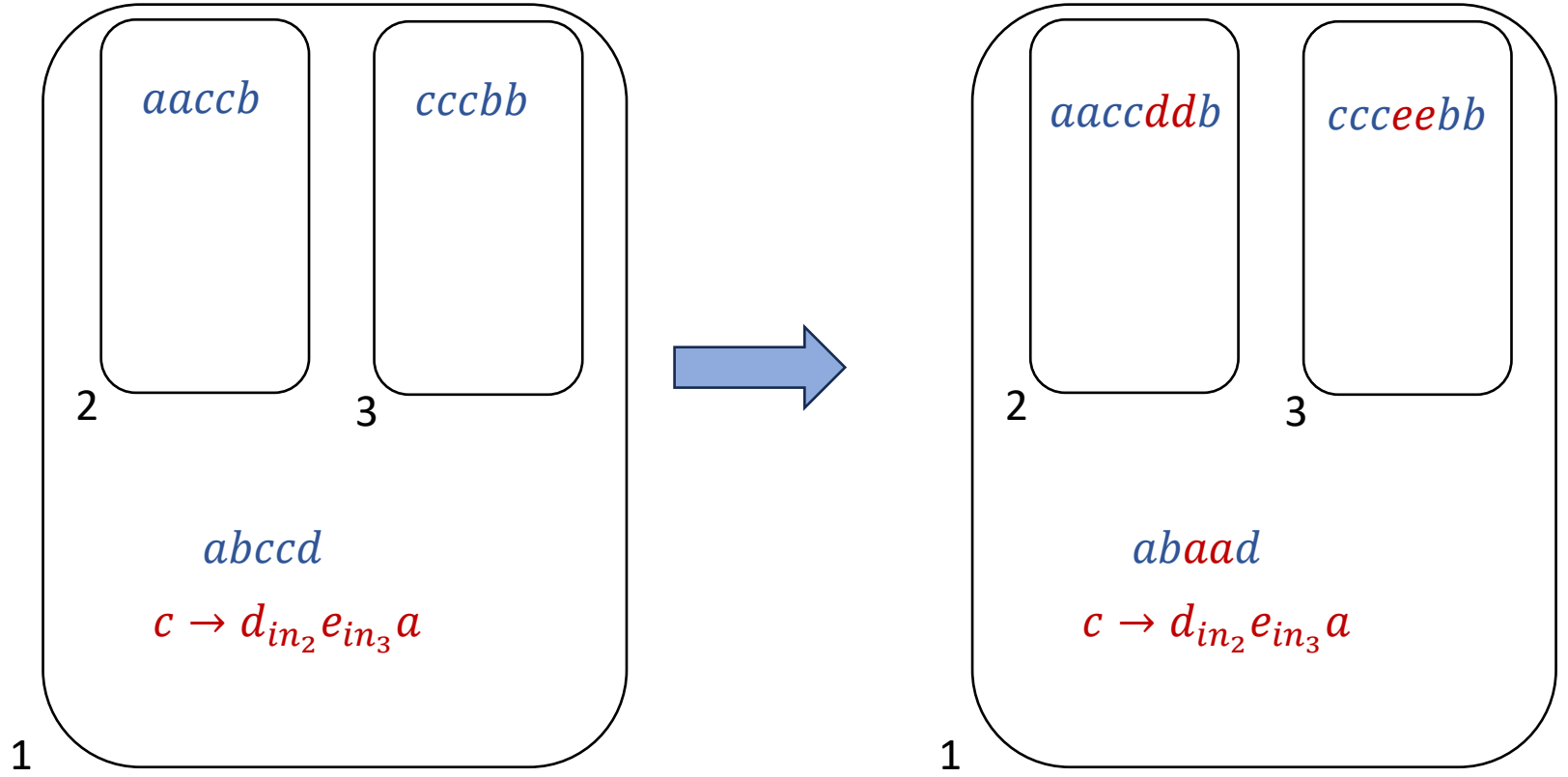
$r_4: accc \to dd$

$r_5: aaacc \to b$

Configuraciones y transiciones

$$\Pi = (O, \mu, w_1, w_2, \dots, w_m, (R_1, \rho_1), (R_2, \rho_2), \dots, (R_m, \rho_m), i_o)$$

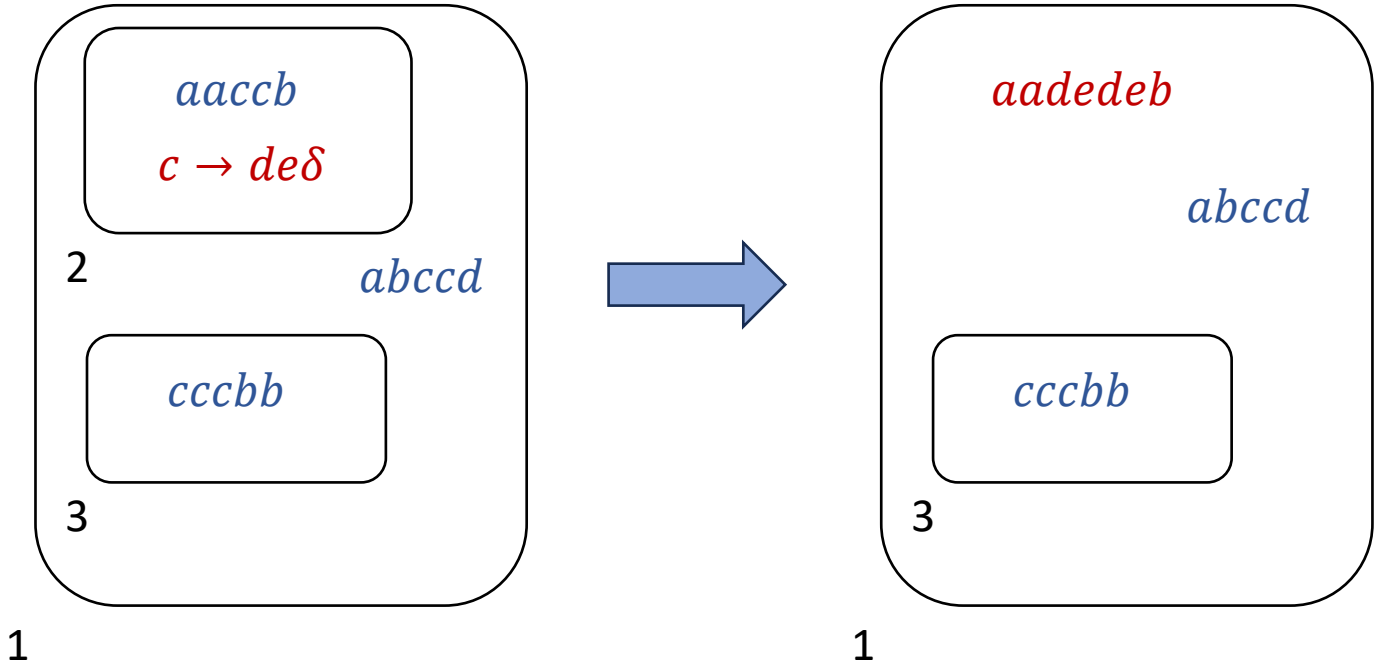
Las reglas con símbolos indexados in_j expulsan los objetos a la región j siempre que sea una región interna adyacente



Configuraciones y transiciones

$$\Pi = (O, \mu, w_1, w_2, \ldots, w_m, (R_1, \rho_1), (R_2, \rho_2), \ldots, (R_m, \rho_m), i_o)$$

Reglas con el símbolo δ hacen que, tras la aplicación de la regla, la membrana correspondiente desaparezca junto con sus reglas y los objetos internos se hereden en la membrana inmediatamente superior de acuerdo con la estructura de membranas. La membrana más externa no se puede disolver.



Lenguajes y conjuntos

Conjunto generado en modo interno $\Pi = (O, \mu, w_1, w_2, \ldots, w_m, (R_1, \rho_1), (R_2, \rho_2), \ldots, (R_m, \rho_m), i_o)$

- Se tienen en cuenta sólo los objetos que se encuentren en la región de salida i_0 cuando el sistema se detiene
- Podemos considerar la proyección de Parikh asociada a los objetos en la región de salida $\Psi_O(w_{i_o})$ o, símplemente, la cantidad de objetos presentes en la región $|w_{i_o}|$

$$N(\Pi) \equiv Ps(\Pi)$$

Lenguaje generado en modo externo $\Pi = (O, \mu, w_1, w_2, \ldots, w_m, (R_1, \rho_1), (R_2, \rho_2), \ldots, (R_m, \rho_m), \infty)$

- Se consideran las cadenas que se forman al enumerar los objetos que se expulsan al entorno exterior en cada paso de computación.
- Si en un paso de computación se expulsan varios objetos, entonces se consideran todas las permutaciones posibles de tales objetos al formar las cadenas.

$$L(\Pi)$$

Catalizadores

- Los sistemas P con catalizadores son un tipo especial de sistemas cooperativos
- En el alfabeto de objetos se distingue un subconjunto de símbolos K que denotan los catalizadores.

$$\Pi = (O, K, \mu, w_1, w_2, \dots, w_m, (R_1, \rho_1), (R_2, \rho_2), \dots, (R_m, \rho_m), i_o)$$

Reglas

$C = \{a_{here}, a_{out}, a_{in_j} : a \in O - K, j \in \{1, .. m\}\}$ (omitiremos el direccionamiento $here$)

- $a \longrightarrow v$ (reglas de evolución sin catalizadores)
- $ca \longrightarrow cv$ (reglas de evolución con catalizadores)

  $a \in O - K \quad v \in C^* \quad c \in K$

Algunas clases de conjuntos definidos por los sistemas P

- Denotaremos por $NOP_m(\alpha, tar)$ a la familia de conjuntos de números naturales $N(\Pi)$ siendo Π un sistema P de grado máximo $m \geq 1$ con reglas de tipo $\alpha \in \{coo, ncoo, cat\}$ y direccionamiento por objetivo (tar).

- Si $m = *$ el grado del sistema no está acotado.

Algunos resultados

- Lema: $(\forall\, m \geq 2)\ NOP_*(\alpha, tar) = NOP_m(\alpha, tar)$
- Lema: $(\forall\, m \geq 1)\ NOP_m(ncoo, tar) \subseteq NOP_m(cat, tar) \subseteq NOP_m(coo, tar)$
- Lema: $NOP_*(ncoo, tar) \subseteq NOP_*(cat, tar) \subseteq NOP_*(coo, tar)$
- Lema: $(\forall m \geq 1)\ NOP_*(coo, tar) = NOP_m(coo, tar) = NRE$ *(completitud de los sistemas P)*
- Lema: $NOP_*(ncoo, tar) = NOP_1(coo) = NCF$

Algunas clases definidas por los sistemas P

- Denotaremos por $NOP_m(\alpha, tar, \delta)$ a la familia de conjuntos de números naturales $N(\Pi)$ siendo Π un sistema P de grado máximo $m \geq 1$ con reglas de tipo $\alpha \in \{coo, ncoo, cat\}$, direccionamiento por objetivo (tar) y disolución de membranas (δ).

- Si $m = *$ el grado del sistema no está acotado.

Lema: $NOP_*(ncoo, tar) \subset NOP_2(ncoo, tar, \delta)$

- Denotaremos por $NOP_m(\alpha, tar, pri)$ a la familia de conjuntos de números naturales $N(\Pi)$ siendo Π un sistema P de grado máximo $m \geq 1$ con reglas de tipo $\alpha \in \{coo, ncoo, cat\}$, direccionamiento por objetivo (tar) y prioridades (pri).

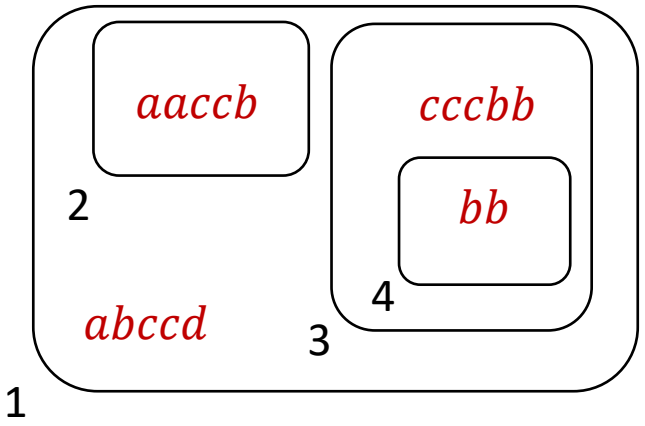
Lema: $NOP_2(cat, tar, pri) = NRE$ *(completitud de los sistemas P)*

Representación de las configuraciones en *Mathematica*

- Una configuración de un sistema P quedará definido por la estructura de membranas y los objetos que contiene cada una de ellas. Por lo tanto, se especifica mediante la región externa.
- Utilizaremos listas anidadas para la representación.

Una región la representaremos de la siguiente forma

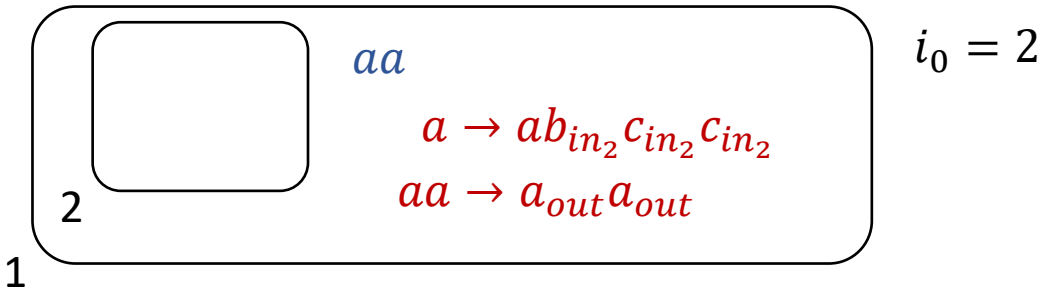
{etiqueta_de_la_región,objetos,regiones_internas}



{1,a,b,c,c,d,{2,a,a,c,c,b},{3,c,c,c,b,b,{4,b,b}}}

Actividades propuestas

1. Diseñe un módulo *Mathematica* que tome como entrada un valor entero n y proporcione como salida el contenido de la región de salida del sistema P que se muestra a continuación después de aplicar n transiciones



2. Diseñe un módulo *Mathematica* que tome como entrada un valor entero n y un valor entero k y proporcione la configuración del sistema P que se muestra a continuación, una vez que el sistema para, indicando regiones y contenidos.