Escuela Técnica Superior de Ingeniería Informática
etsinf

DSIC
DEPARTAMENTO DE SISTEMAS
INFORMÁTICOS Y COMPUTACIÓN

Computabilidad y Complejidad

Práctica 3: Computación con membranas. Sistemas P

Computación con membranas. Sistemas P

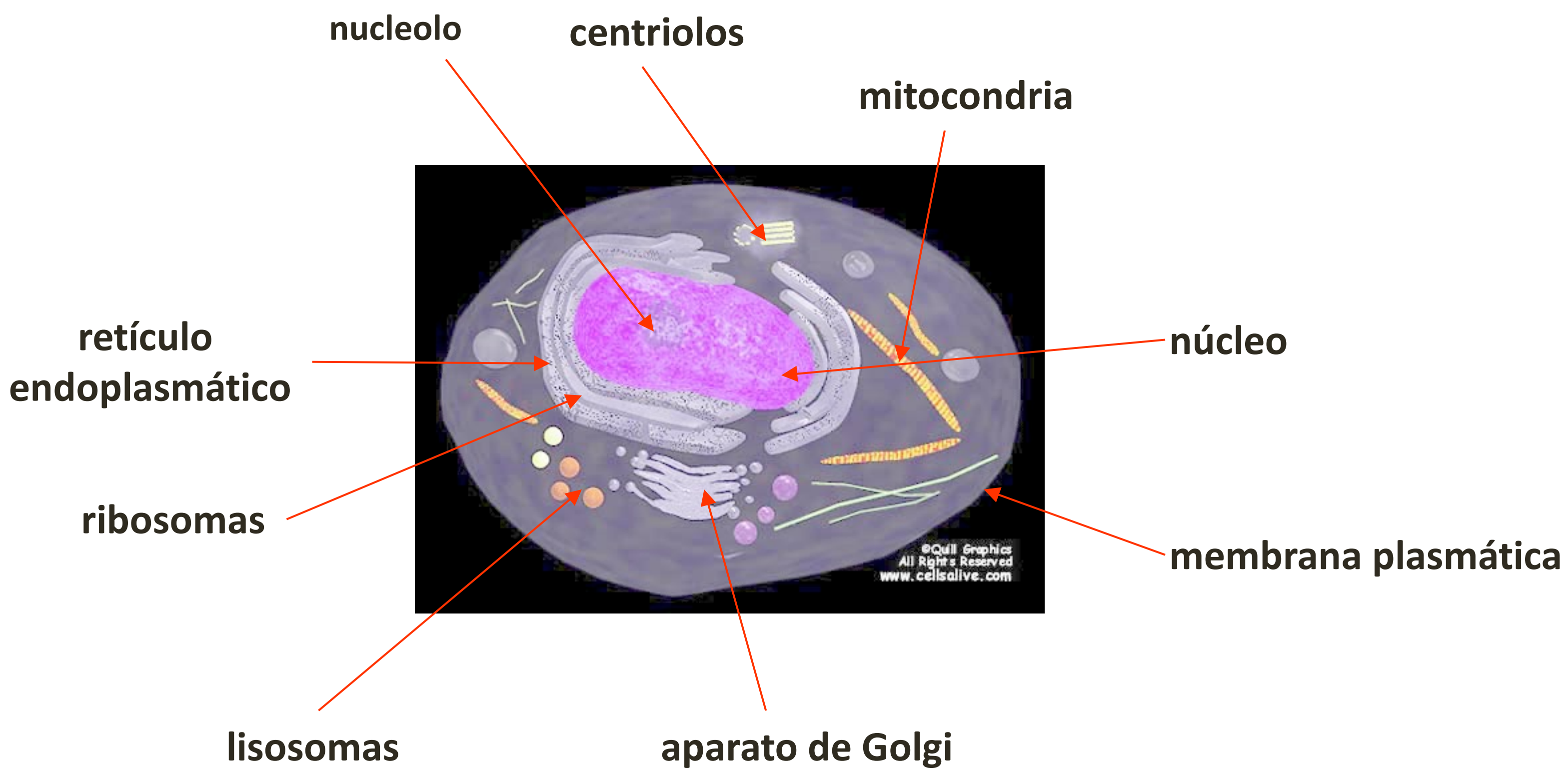
Índice:

1. La célula.
2. Multiconjuntos.
3. Proyecciones de Parikh y conjuntos naturales.
4. Sistemas P. Sistemas P de transición.
5. Lenguajes y conjuntos.
6. Algunas clases definidas por los Sistemas P.
7. Actividades propuestas

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



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

Multiconjuntos

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

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

- $M(a)$ es la ***multiciplicidad*** del elemento a
- $supp(M) = \{ a \in V : M(a) > 0\}$ 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 $a_1^{M(a1)}\, a_2^{M(a2)} \ldots a_n^{M(an)}$ o cualquiera de sus permutaciones

Proyecciones de Parikh y conjuntos naturales

Dado un alfabeto $\Sigma = \{a_1, a_2, \ldots, a_n\}$ y una cadena $x = x_1 x_2 \ldots x_m \in \Sigma^*$ se define La proyección de Parikh de x en Σ como la aplicación $\Psi_\Sigma : \Sigma^* \rightarrow \mathrm{N}^n$

$$\Psi_\Sigma(x_1 x_2 \ldots x_m) = (|x|_{a1}, |x|_{a2}, \ldots, |x|_{an})$$

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

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 $\Sigma = \{a_1, a_2, \ldots, a_n\}$ y un lenguaje $L \subseteq \Sigma^*$ se define el conjunto de naturales de L como NL = $\{ n \in \mathrm{N} : x \in L \wedge |x| = n \}$. Podemos definir el conjunto NL a partir de PsL (como suma de las componentes de los vectores)

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

$$\boxed{\mathrm{NFIN} \subset \mathrm{NREG} = \mathrm{NLIN} = \mathrm{NCF} \subset \mathrm{NCS} \subset \mathrm{NRE}}$$

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

Un ***sistema P de transición*** de grado m se define como la tupla

$$\Pi = (O, \mu, w_1, \ldots, w_m , (R_1,\rho_1), \ldots, (R_m,\rho_m), i_0)$$

O alfabeto de objetos[1]
μ estructura de membranas (representable mediante un árbol)
$w_1, \ldots, w_m$ cadenas asociadas a cada membrana (multiconjuntos)
$(R_1,\rho_1), \ldots, (R_m,\rho_m)$ reglas de evolución y relaciones de prioridad
$i_0 \in \{ 1, \ldots, m \} \cup \{ \infty \}$ membrana de salida

[1] En algunas variantes de sistemas P de transición, el alfabeto O se divide en tres alfabetos: objetos, *catalizadores* y objetos de salida.
Nosotros en la presente práctica no haremos distinción entre los objetos.

Sistemas P de transición

$\Pi = (\, O, \mu, w_1, \ldots, w_m, (R_1,\rho_1), \ldots, (R_m,\rho_m), i_0 \,)$

$(R_1,\rho_1), \ldots, (R_m,\rho_m)$ reglas de evolución y relaciones de prioridad

R_i es un conjunto finito de reglas de evolución

$$(u,v) \quad \text{o} \quad u \rightarrow v$$

donde $u \in O^*$ y $v = v'$ o $v = v'\delta$ con $v' \in Coop^*$ y $\delta \notin O$

$$Coop = \{\, a_{here}, a_{out}, a_{inj} \mid a \in O\,,\, 1 \leq j \leq m \,\}$$

in_j denota *direccionamiento por objetivo*

El símbolo δ denota la disolución de la membrana

Dada la regla $u \rightarrow v$ llamaremos ***radio*** de la regla a $|u|$

ρ_i es un orden parcial sobre R_i (***prioridad***)

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

Sistemas P de transición

(*representación mediante diagramas*)

$\Pi = (O, \mu, w_1, \ldots, w_m , (R_1,\rho_1), \ldots, (R_m,\rho_m), i_0)$

Ejemplo

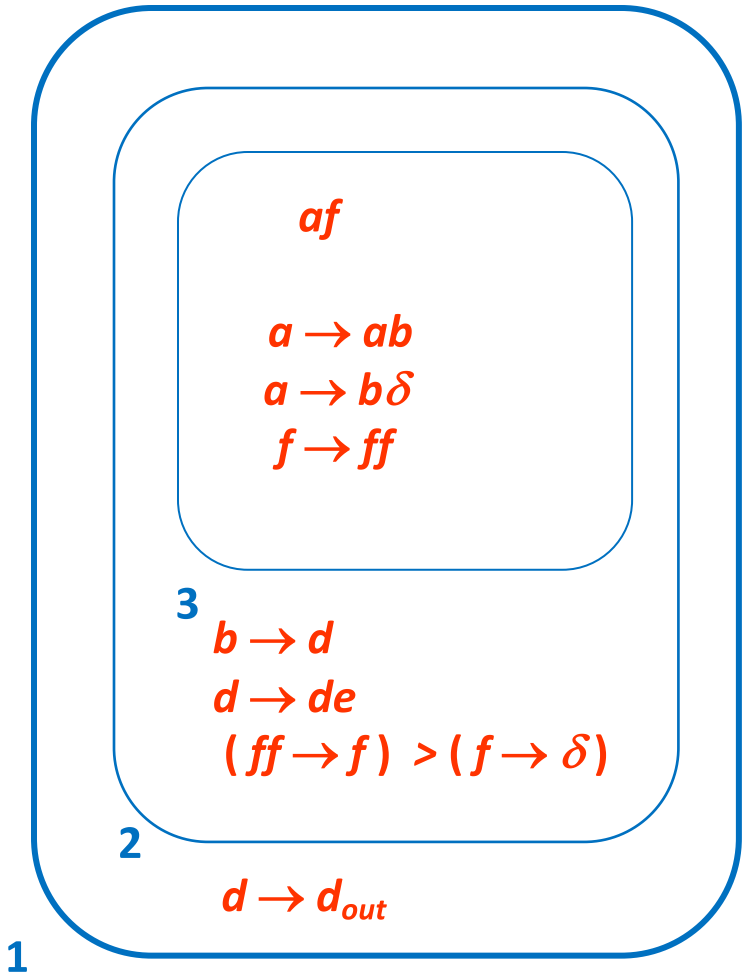
$O = \{ a,b,d,e,f \} \qquad i_0 = 1$

$\mu = [_1 [_2 [_3]_3]_2]_1$

$w_1 = \lambda \quad R_1 = \{ d \to d_{out}\} \qquad \rho_1 = \emptyset$

$w_2 = \lambda \quad R_2 = \{ b \to d,\ d \to de,\ r_1: ff \to f ,\ r_2 : f \to \delta\} \quad \rho_2 = \{ r_1 > r_2 \}$

$w_3 = af \quad R_3 = \{ a \to ab,\ a \to b\delta,\ f \to ff \} \qquad \rho_3 = \emptyset$


af
a → ab
a → bδ
f → ff
3
b → d
d → de
(ff → f) > (f → δ)
2
d → d_out
1

Configuraciones, transiciones, lenguajes y conjuntos de los sistemas P (I)

$\Pi = (O, \mu, w_1, \ldots, w_m, (R_1,\rho_1), \ldots, (R_m,\rho_m), i_0)$

$(\mu', w_{i1}, \ldots, w_{ik})$ *configuración de* Π
$(\mu, w_1, \ldots, w_m)$ *configuración inicial de* Π

Transiciones

$$C_1 = (\mu', w_{i1}, \ldots, w_{ik}) \quad C_2 = (\mu'', w'_{j1}, \ldots, w'_{jl})$$

$$C_1 \Rightarrow C_2$$

Configuraciones, transiciones, lenguajes y conjuntos de los sistemas P (II)

Aplicación de las reglas de evolución

Todas las reglas que se puedan aplicar en una región se aplican en paralelo

$aaaccb$

$a \rightarrow d \quad a \rightarrow f$

$aa \rightarrow bc$

$$\Downarrow$$

$dddccb$	$fffccb$	$bcaccb$	...
$a \rightarrow d \quad a \rightarrow f$	$a \rightarrow d \quad a \rightarrow f$	$a \rightarrow d \quad a \rightarrow f$	
$aa \rightarrow bc$	$aa \rightarrow bc$	$aa \rightarrow bc$	

Configuraciones, transiciones, lenguajes y conjuntos de los sistemas P (III)

Aplicación de las reglas de evolución

La prioridad entre reglas puede impedir que una regla se aplique

$aaccb$ $r_1 : (a \to b) >$ $r_2 : (aa \to bc)$ $r_3 : (b \to f)$	$\Rightarrow$	$bbccb$ $r_1 : (a \to b) >$ $r_2 : (aa \to bc)$ $r_3 : (b \to f)$

Una interpretación menos rígida permite que se apliquen reglas con menor prioridad si es posible y siempre que se hayan aplicado todas las reglas más prioritarias. (en nuestro ejemplo obtendríamos el multiconjunto *bbccf*)

Configuraciones, transiciones, lenguajes y conjuntos de los sistemas P (IV)

Aplicación de las reglas de evolución

Reglas con símbolos a_{here} hacen que los símbolos permanezcan en la región

$aaccb$

$a \rightarrow bb$

$\Rightarrow$

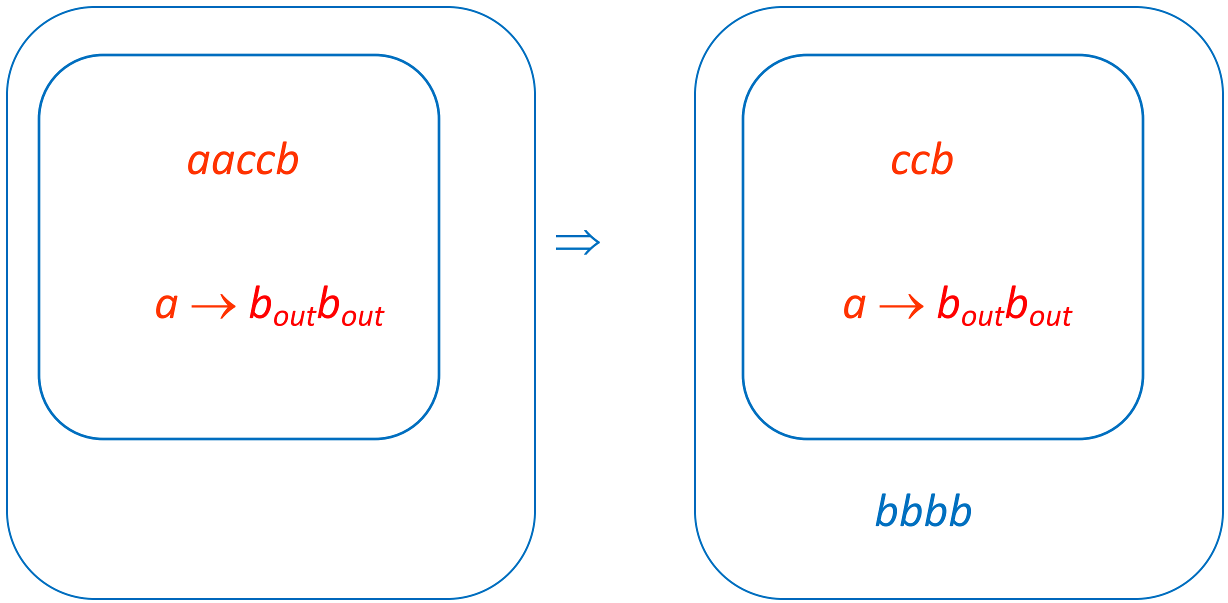
$bbbbccb$

$a \rightarrow bb$

Configuraciones, transiciones, lenguajes y conjuntos de los sistemas P (V)

Aplicación de las reglas de evolución

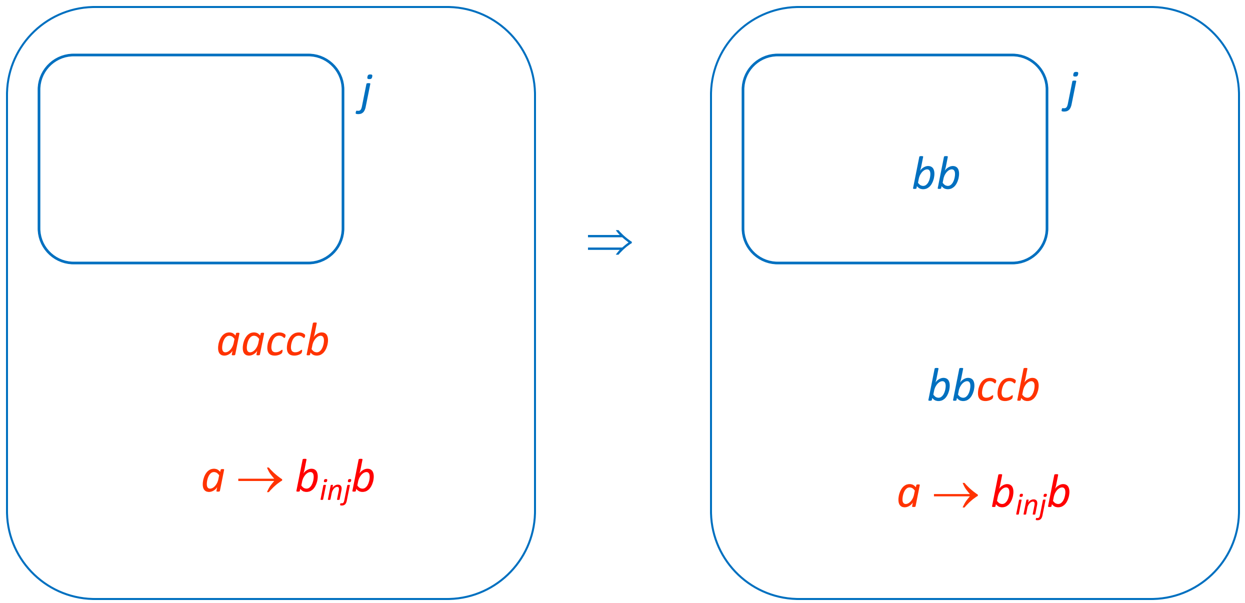
Reglas con símbolos a_{out} hacen que los símbolos pasen a la región inmediatamente superior de acuerdo con μ (o salgan del sistema)



Configuraciones, transiciones, lenguajes y conjuntos de los sistemas P (VI)

Aplicación de las reglas de evolución

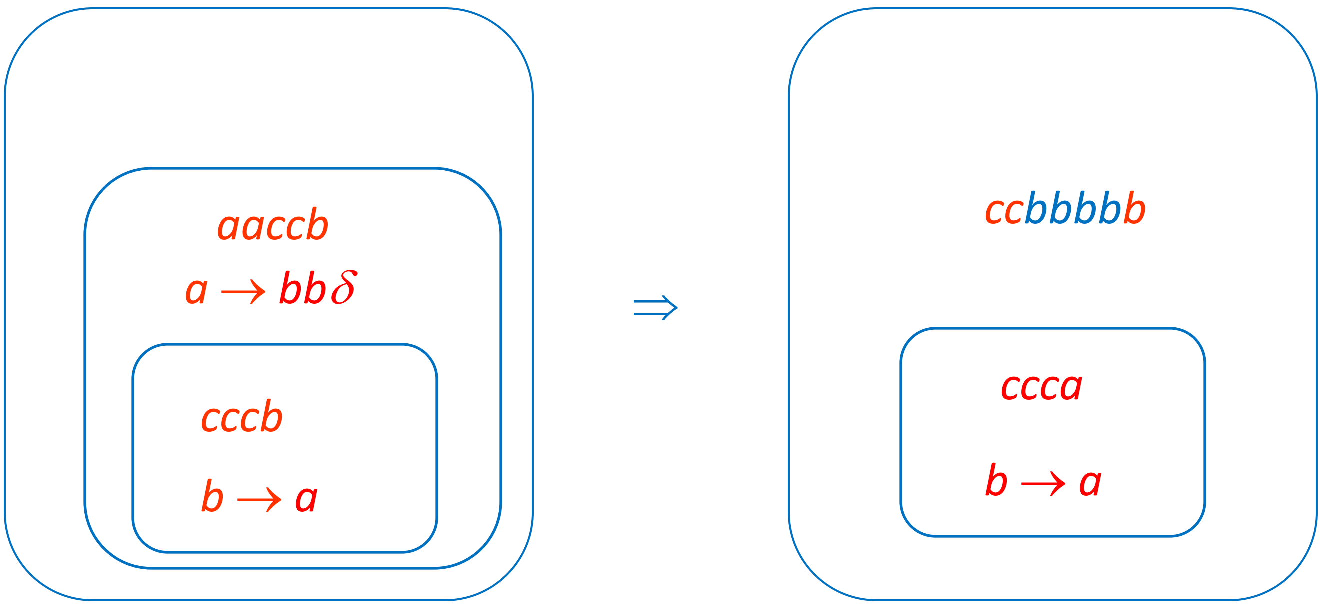
Reglas con símbolos a_{inj} hacen que los símbolos pasen a la región j siempre que j sea una región adyacente inferior de acuerdo con μ



Configuraciones, transiciones, lenguajes y conjuntos de los sistemas P (VII)

Aplicación de las reglas de evolución

Reglas con el símbolo δ hacen que, tras la aplicación de la regla, la membrana correspondiente desaparezca y los objetos internos se hereden en la membrana inmediatamente superior de acuerdo con μ



Configuraciones, transiciones, lenguajes y conjuntos de los sistemas P (VIII)

Conjunto generado en <u>modo interno</u>

$$\Pi = (O, \mu, w_1, \ldots, w_m, (R_1,\rho_1), \ldots, (R_m,\rho_m), i_0)$$

$\Psi_O(w)$ (Proyección de Parikh asociada a O)

$Ps(\Pi)$ (Conjunto de Parikh) $N(\Pi) \equiv Ps(\Pi)$

(en este caso no haremos distinción entre las proyecciones de Parikh y los conjuntos de naturales que inducen)

Lenguaje generado en <u>modo externo</u>

$$\Pi = (O, \mu, w_1, \ldots, w_m, (R_1,\rho_1), \ldots, (R_m,\rho_m), \infty)$$

$L(\Pi)$ Cadenas recogidas desde el exterior
(incluyendo permutaciones cuando se expulsan varios objetos)

Algunas clases definidas por los sistemas P

***Denotaremos por $NOP_m(\alpha$, tar) a la familia de conjuntos de naturales* N(Π) *siendo Π un sistema P de grado máximo $m \geq 1$ con reglas de tipo α y direccionamiento por objetivo.* $\alpha \in$ {*coo, ncoo, cat*[1]} (si m=* el grado del sistema no está acotado en grado)**

<u>Lema</u> $NOP_*(\alpha, tar) = NOP_m(\alpha, tar)$ $m \geq 2$

<u>Lema</u> $NOP_m(ncoo, tar) \subseteq NOP_m(cat, tar) \subseteq NOP_m(coo, tar)$ $m \geq 1$

<u>Lema</u> $NOP_*(ncoo, tar) \subseteq NOP_*(cat, tar) \subseteq NOP_*(coo, tar)$

<u>Lema</u> $NOP_*(coo, tar) = NOP_m(coo, tar) = NRE$ $m \geq 1$

<u>Lema</u> $NOP_*(ncoo, tar) = NOP_1(ncoo) = NCF$

[1] cat denota que el sistema utiliza catalizadores que son objetos que no se transforman en ninguna regla

Algunas clases definidas por los sistemas P

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

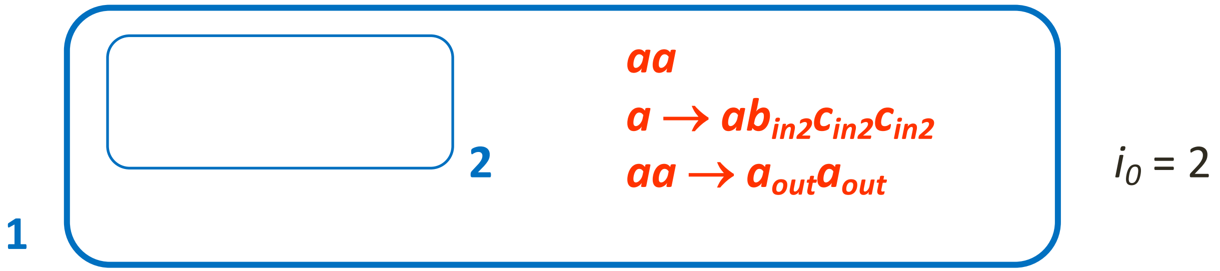
Lema $NOP_*(ncoo, tar) \subset NOP_2(ncoo, tar, \delta)$

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

Lema. $NOP_2(cat, tar, pri) = NRE$

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 como salida la configuración de salida (indicando regiones y contenidos) del sistema P que se muestra a continuación una vez que ya no se puedan aplicar más reglas