MIARFID Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

José Javier Calvo Moratilla
2021/2022

Computación Natural
Ejercicios propuestos - Computación biomolecular

1. Defina un algoritmo de tubos basado en la técnica de Lipton que resuelva la siguiente instancia del SAT

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3)$$

1) $input(N_0)$
2) $N_1 = S(N_0, 1, 1)$
3) $N'_1 = \overline{S}(N_0, 1, 1)$
4) $N_2 = S(N'_1, 2, 0)$
5) $N_3 = merge(N_1, N_2)$
6) $N_4 = S(N_3, 1, 0)$
7) $N'_4 = \overline{S}(N_3, 1, 0)$
8) $N_5 = S(N'_4, 3, 0)$
9) $N_6 = merge(N_4, N_5)$
10) $N_7 = S(N_6, 2, 0)$
11) $N'_7 = \overline{S}(N_6, 2, 0)$
12) $N_8 = S(N'_7, 3, 1)$
13) $N_9 = merge(N_7, N_8)$
14) $Detect(N_9)$

2. Proporcione un sistema de stickers para el lenguaje

$$\{ww^r : w \in \{a,b\}^*\}$$

Clasifique el sistema en función de la forma de sus *dominoes*. ¿cómo es el lenguaje de sus computaciones primitivas? ¿cómo es el lenguaje de sus computaciones con retardo acotado?

Se define un sistema de stickers ρ:

$V \; = \; \{a, b\}$
$\gamma \; = \; \{\{a, a\}, \{b, b\}\}$
$A \; = \; \{[aa, \; aa]\}$
$D \; = \; \{ \; ((a\,a), (a\,a)), ((b\,b), (b\,b))\}$

Lenguaje arbitrario:
$L_n = LM_n(\rho) \; = \; \{[W\,W][W\,W]^r : \; W \; \in \{a, b\} \; *\}$

Después de realizar las operaciones de stickers la molécula resultante crece tanto por la derecha como por la izquierda, por ello no se puede considerar unilateral, regular ni simple.

El lenguaje generado por ϱ mediante computaciones primitivas:
$L_p(\rho) \; = \; L_n(\rho)$

Con los dominós disponibles en el ejemplo no se pueden generar moléculas no completas, así que podemos considerar el lenguaje generado por computaciones primitivas igual que el lenguaje arbitrario.

Para identificar el retardo las operaciones de dominós siempre provocan un retardo 'd' igual o mayor a cero:
$\forall d \geq 0, \; L_d(\rho) \; = \; L_p(\rho)$

3. Proporcione un sistema de stickers ρ de forma que

$$L_n(\rho) \cap a^*b^*c^* = \{a^{n+1}b^n c^n : n \geq 1\}.$$

Clasifique el sistema en función de la forma de sus *dominoes*. ¿cómo es el lenguaje de sus computaciones primitivas? ¿cómo es el lenguaje de sus computaciones con retardo acotado?

Se define un sistema de sticker ρ:

$V = \{a, b, c\}$
$\gamma = \{\{a, a\}, \{b, b\}, \{c, c\}\}$
$A = \{[a, a]\}$
$D = \{ ((a\,\lambda), (b\,\lambda)), ((\lambda\,\lambda), (c\,\lambda)), ((\lambda\,\lambda), (\lambda\,b)), ((\lambda\,a), (\lambda\,c)) \}$

Después de realizar las operaciones de stickers la molécula resultante crece tanto por la derecha como por la izquierda, por ello no se puede considerar unilateral, regular ni simple, igual que en el ejercicio anterior.

El lenguaje generado por ϱ mediante computaciones primitivas puede generar moléculas no completas, así que diferencia con el ejercicio anterior el lenguaje de generado mediante computaciones primitivas es un subconjunto del lenguaje arbitrario generado.
$L_p(\rho) \subseteq L_n(\rho)$

Las operaciones de dominós siempre provocan un retardo 'd' igual o mayor a uno y menor que n:
$L_d(\rho) = L_n(\rho) \; 1 \leq d \leq n$

4. Proporcione un AFWK para el lenguaje de hebra superior

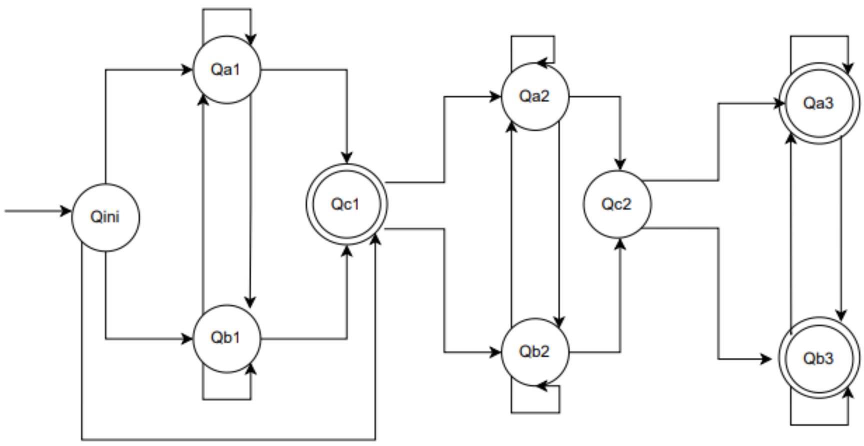
$$\{wcw : w \in \{a, b\}^*\}$$

Clasifique el autómata en función de sus transiciones y del conjunto de estados y estados finales.

Para la resolución del ejercicio se emplea un AFWK de tipo simple en el que se puede consumir un símbolo de la hebra superior o inferior por la izquierda.

Se definen los siguientes parámetros:

$V \ = \ \{a, b, c\}$
$K \ = \ \{Q_{ini}, Q_{a1}, Q_{b1}\ Q_{c1}, Q_{a2}, Q_{b2}\ Q_{c2}, Q_{a3}, Q_{b3}\}$
$\gamma \ = \ \{\{a, a\}, \{b, b\}, \{c, c\}\}$
$S_0 = \ Q_{ini}$
$F \ = \{Q_{c1}, Q_{a3}, Q_{b3}\}$



Se observan las siguientes transiciones:

Qini	Qa1	Qb1	Qc1
$\delta(Q_{ini}, (a\,\lambda)) \ = \ Q_{a1}$ $\delta(Q_{ini}, (b\,\lambda)) \ = \ Q_{b1}$ $\delta(Q_{ini}, (c\,c)) \ = \ Q_{c1}$	$\delta(Q_{a1}, (a\,\lambda)) \ = \ Q_{a1}$ $\delta(Q_{a1}, (b\,\lambda)) \ = \ Q_{b1}$ $\delta(Q_{a1}, (c\,\lambda)) \ = \ Q_{c1}$	$\delta(Q_{b1}, (b\,\lambda)) \ = \ Q_{b1}$ $\delta(Q_{b1}, (a\,\lambda)) \ = \ Q_{a1}$ $\delta(Q_{b1}, (c\,\lambda)) \ = \ Q_{c1}$	$\delta(Q_{c1}, (a\,a)) \ = \ Q_{a2}$ $\delta(Q_{c1}, (b\,b)) \ = \ Q_{b2}$

Qa2	Qb2	Qc2	Qa3
$\delta(Q_{a2}, (a\ a)) = Q_{a2}$ $\delta(Q_{a2}, (b\ b)) = Q_{b2}$ $\delta(Q_{a2}, (\lambda\ c)) = Q_{c2}$	$\delta(Q_{b2}, (b\ b)) = Q_{b2}$ $\delta(Q_{b2}, (a\ a)) = Q_{a2}$ $\delta(Q_{b2}, (\lambda\ c)) = Q_{c2}$	$\delta(Q_{c2}, (\lambda\ a)) = Q_{a2}$ $\delta(Q_{c2}, (\lambda\ b)) = Q_{b2}$	$\delta(Q_{a3}, (\lambda\ b)) = Q_{b3}$ $\delta(Q_{a3}, (\lambda\ a)) = Q_{a3}$
Qb3			
$\delta(Q_{b3}, (\lambda\ a)) = Q_{a3}$ $\delta(Q_{b3}, (\lambda\ b)) = Q_{b3}$			

El AFWK propuesto se considera de tipo 1-limitado $1WK(\alpha)$, dado que solo se consume un símbolo por cada hebra en cada transición.

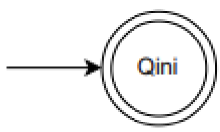
También, observando el número de estados normales y finales, se considera que el AFWK no es de tipo stateless y tampoco all-final.

> 5. Proporcione un AFWK en cualquiera de sus variantes para el lenguaje
>
> $$\{ww^r : w \in \{a, b\}^*\}$$
>
> Clasifique el autómata en función de sus transiciones y del conjunto de estados y estados finales.

A diferencia de AFWK del ejercicio número cuatro, se emplea un AFWK de tipo "reverso", en el cual en cada transición se consume un símbolo en la hebra superior por la izquierda y otro símbolo en la hebra inferior por la derecha.

$V = \{a, b\}$
$K = \{Q_{ini}\}$
$\gamma = \{\{a, a\}, \{b, b\}\}$
$S_0 = Q_{ini}$
$F = Q_{ini}$

El autómata se representa con un único estado inicial, que también es final.



Se observan las siguientes transiciones:

Qini
$\delta(Q_{ini}, (a\ b)) = Q_{ini}$ $\delta(Q_{ini}, (b\ b)) = Q_{ini}$

El número de estados es igual uno y el estado inicial es igual al final, por ello observando los estados se considera un AFWK de tipo "all-final".
Las transiciones consumen solo un símbolo en cada hebra, por tanto el AFWK se considera de tipo 1-limitado $1WK(\alpha)$.

Como conclusión el lenguaje reconocido por el AFWK es de tipo "all-final" + "1-limitado" $F1WK(\alpha)$.

6. Dado el lenguaje $L = \{abba, aba, aabbabba\}$ establezca si la cadena $ababbabbaaabba$ pertenece a $D^*(L)$. Justifique su respuesta

Para poder demostrar que la cadena ababbabbaaabbaa pertenece a D*(L) se realizan operaciones de duplicación, partiendo de "abba", perteneciente al lenguaje inicial.

Se realizan las siguientes operaciones de duplicación:

ab **|** ba → ab | abba → ababba | abba → ababbaa | abba → ababba | bbaaabba → ababbabbaaabba

Después de haber aplicado las operaciones se demuestra que ababbabbaaabba pertenece a D*(L)

7. Dado el esquema de splicing $\sigma = (V, R)$ donde R contiene únicamente la regla caba#a$cab#a y dado el lenguaje L={cabb,cabab,cabaab}, obtenga $\sigma_1^*(L)$.

Para la resolución del problema se utilizan las reglas de "splicing". Si se observa "caba#a$cab#a" la primera parte de la operación anterior al símbolo "$" corresponden a la hebra superior, la segunda parte a la derecha del símbolo a la hebra inferior.

La operación se realiza justamente dónde aparece el símbolo "#", siendo la cadena que aparece a la derecha la que primero se intercambia entre las hebras y por último quedándote solo con el resultado de la hebra superior.

Inicialmente se tiene un lenguaje L que se puede entender como un conjunto de tres elementos {cabb, cabab, cabaab}, al que se le van aplicando operaciones de splicing de manera consecutiva, para así obtener un lenguaje resultante.

$(cabab,\ cabaab) \vdash_r cabaab$ $cabaab \in \sigma_1^1$
$(cabaab,\ cabaaab) \vdash_r cabaaaab$ $cabaaaab \in \sigma_1^2$
$(cabaab,\ cabaaaab) \vdash_r cabaaaaab$ $cabaaaaab \in \sigma_1^3$
$(cabaab,\ cabaaaaab) \vdash_r cabaaaaaab$ $cabaaaaaab \in \sigma_1^4$
Obteniendo el siguiente lenguaje: $\sigma_1^* = \{caba^n b,\ n \geq\ 0\}$