

Un algoritmo lineal para el conteo de #2SAT en fórmulas tipo árbol con subfórmulas serial paralelo

López Medina, M. A.; Marcial Romero, J. R.; González Ruiz, J. L.; Hernández Servín, J. A.

Información del Artículo	Resumen
Recibido: 11-mar-2023 Aceptado: 27-jun-2023	Este artículo presenta un algoritmo de tiempo lineal para resolver el problema de conteo #2SAT en fórmulas booleanas de dos formas normales conjuntivas (2-CNF) con una estructura específica. El enfoque se centra en fórmulas cuya representación gráfica forma árboles con subgrafos serie-paralelo en sus nodos interiores u hojas.
Palabras clave: 2SAT, teoría de la complejidad, teoría de grafos, algoritmos lineales, fórmulas tipo árbol,	

1. Introducción

SAT(F) es el problema de decisión que consiste en determinar si la fórmula Booleana F tiene una asignación de valores de verdad sobre las variables de F, que al ser evaluada utilizando la lógica Booleana clásica de como resultado verdadero. Si la fórmula F está en dos forma normal conjuntiva entonces SAT(F) puede resolverse en tiempo polinomial, sin embargo si F está en k forma normal conjuntiva, $k > 2$, entonces SAT(F) es un problema NP-Completo. #SAT(F) es la versión de conteo del problema SAT, que consiste en determinar el número de asignaciones sobre las variables de F que al ser evaluadas dan como resultado verdadero. #SAT(F) pertenece a la clase de problemas #P-Completo incluso cuando F está en dos forma normal conjuntiva (#2SAT) [1].

Aunque el problema #2SAT es #P-Completo, hay casos que pueden resolverse en tiempo polinomial [2] [3]. Por ejemplo, si la representación gráfica de la fórmula es acíclica, entonces el número de modelos se puede calcular en tiempo lineal. Actualmente los métodos para resolver #2SAT se basan en descomponer la fórmula a través de sus variables o cláusulas para obtener fórmulas más simples. El algoritmo reportado con mejor complejidad en tiempo fue presentado por Wahlström [4], la cual es $O(1,2377^n)$ (dónde n es el número de variables en la fórmula).

Sobre gráficas serial paralelo el trabajo más cercano al conteo de modelos está relacionado con el reconocimiento de este tipo de gráficas y algunos problemas de decisión.

Schoenmakers [5] muestra un algoritmo de tiempo lineal para reconocimiento de gráficas serial paralelo, calculando una representación *origen-sumidero* de esta gráfica usando un árbol de expansión construido por anchura. La complejidad de construir esta representación es $O(n\sqrt{n})$. Así mismo Eppstein [6] propone un algoritmo para reconocer gráficas serial paralelo dirigidas y no dirigidas, basado en una caracterización llamada *descomposición en oreja*. Por otro lado Takamizawa [7] dice que si una gráfica es restringida a la clase serial paralelo, entonces existen algoritmos de tiempo lineal para problemas de decisión y combinatorios como: cubierta

de vértices mínima, conjunto de vértices independientes maximal, subgráfica de línea máxima (inducida), subgráfica outerplanar máxima (inducida), conjunto mínimo de vértices de *retro alimentación*, subgráfica máxima en escalera (inducida), cobertura de trayectoria mínima, entre otros.

En este artículo se presenta un algoritmo para el conteo de modelos en árboles de gráficas serial paralelo, el cual tiene una complejidad lineal.

2. Metodología

Preliminares

En esta sección se describen conceptos básicos utilizados en el desarrollo del artículo.

Forma Normal Conjuntiva

Siendo $X = \{x_1, \dots, x_n\}$ un conjunto de n variables Booleanas (que sólo pueden tener dos posibles valores falso (0) o verdadero (1)). Una literal puede ser la variable x_i , denotada también como x_i^1 , o su negación $\bar{x}_i$, denotada como x_i^0 .

Una fórmula en forma normal conjuntiva es una conjunción de cláusulas, y las cláusulas son una disyunción de variables.

Una asignación s sobre F es una función Booleana $s : v(F) \rightarrow \{0, 1\}$ y definida como:

$$s(x^0) = 1 \text{ si } s(x^1) = 0, \text{ de otra forma, } s(x^0) = 0.$$

Esta asignación puede usarse con conjunciones y disyunciones:

- $s(x \wedge y) = 1$ si $s(x) = s(y) = 1$, de otra forma, $s(x \wedge y) = 0$
- $s(x \vee y) = 0$ si $s(x) = s(y) = 0$, de otra forma, $s(x \vee y) = 1$

Sea F una fórmula Booleana en forma normal conjuntiva, y s una asignación sobre las variables de F , s satisface a F si se cumple que para toda cláusula c en F , $s(c) = 1$.

La gráfica restringida de una 2 Forma Normal Conjuntiva

Existen algunas representaciones gráficas de una forma normal conjuntiva, en este caso se describe la gráfica primal signada (gráfica restringida). Sea F una 2 forma normal conjuntiva, su gráfica restringida se denota por $G_F = (V(F), E(F))$, donde los vértices de la gráfica son las variables ($V(F) = v(F)$) y el conjunto de aristas $E(F)$ se conforma por todas las cláusulas en F , esto es para cada cláusula $\{x_i^\epsilon, x_j^\gamma\} \in F$ existe la arista $\{x_i^\epsilon, x_j^\gamma\} \in E(F)$. Para $x \in V(F)$, $\delta(x)$ denota su grado, el número de aristas incidentes en x . Cada arista $\{x_i^\epsilon, x_j^\gamma\}$ tiene asociado un par (ϵ, γ) , que representa cuando las variables x_i o x_j aparecen negadas o no.

Métodos de conteo para #2SAT

En artículos relacionados al conteo de modelos sobre fórmulas en dos forma normal conjuntiva se considera como idea básica el uso de una tupla (α_i, β_i) sobre cada vértice x_i en la gráfica G , dónde α_i representa el número de veces que x_i aparece con una asignación positiva en los modelos de G y β_i el número de veces que aparece negaiva.

Existen métodos reportados para el conteo de modelos de una dos forma normal conjuntiva F [8], por ejemplo:

Si la gráfica representa un camino de la forma $P_n = \{\{x_1^{\epsilon_1}, x_2^{\gamma_1}\}, \{x_2^{\epsilon_2}, x_3^{\gamma_2}\}, \dots, \{x_{n-1}^{\epsilon_{n-1}}, x_n^{\gamma_{n-1}}\}\}$ de n vértices, el número de modelos está dado por la suma de los elementos (α_n, β_n) . Donde $(\alpha_1, \beta_1) = (1, 1)$ y la tupla para los demás vértices se calcula según los signos de acuerdo a la siguiente recurrencia:

$$(\alpha_i, \beta_i) = \begin{cases} (\alpha_{i-1} + \beta_{i-1}, \alpha_{i-1}) & \text{if } (\epsilon_{i-1}, \gamma_{i-1}) = (1, 1) \\ (\alpha_{i-1}, \alpha_{i-1} + \beta_{i-1}) & \text{if } (\epsilon_{i-1}, \gamma_{i-1}) = (1, 0) \\ (\alpha_{i-1} + \beta_{i-1}, \beta_{i-1}) & \text{if } (\epsilon_{i-1}, \gamma_{i-1}) = (0, 1) \\ (\beta_{i-1}, \alpha_{i-1} + \beta_{i-1}) & \text{if } (\epsilon_{i-1}, \gamma_{i-1}) = (0, 0) \end{cases} \quad (1)$$

Si $\{x_i^{\epsilon_1}, x_j^{\gamma_1}\}$ y $\{x_i^{\epsilon_2}, x_j^{\gamma_2}\}$ son dos cláusulas paralelas en la fórmula F , el número de modelos para estas cláusulas es dado por:

$$(\alpha_j, \beta_j) = \begin{cases} (\alpha_i, \alpha_i) & \text{if } (\epsilon_1, \gamma_1) = (1, 1) \text{ and } (\epsilon_2, \gamma_2) = (1, 0) \\ (\alpha_i + \beta_i, 0) & \text{if } (\epsilon_1, \gamma_1) = (1, 1) \text{ and } (\epsilon_2, \gamma_2) = (0, 1) \\ (\beta_i, \alpha_i) & \text{if } (\epsilon_1, \gamma_1) = (1, 1) \text{ and } (\epsilon_2, \gamma_2) = (0, 0) \\ (\alpha_i, \beta_i) & \text{if } (\epsilon_1, \gamma_1) = (1, 0) \text{ and } (\epsilon_2, \gamma_2) = (0, 1) \\ (0, \alpha_i + \beta_i) & \text{if } (\epsilon_1, \gamma_1) = (1, 0) \text{ and } (\epsilon_2, \gamma_2) = (0, 0) \\ (\beta_i, \beta_i) & \text{if } (\epsilon_1, \gamma_1) = (0, 1) \text{ and } (\epsilon_2, \gamma_2) = (0, 0) \end{cases} \quad (2)$$

Si $\{x_i^{\epsilon_1}, x_j^{\gamma_1}\}$, $\{x_i^{\epsilon_2}, x_j^{\gamma_2}\}$ y $\{x_i^{\epsilon_3}, x_j^{\gamma_3}\}$ son tres cláusulas paralelas en la fórmula F , el número de modelos para las cláusulas se calcula:

$$(\alpha_j, \beta_j) = \begin{cases} (0, \alpha_i) & \text{if } (\epsilon_1, \gamma_1) = (1, 1) \text{ and } (\epsilon_2, \gamma_2) = (1, 0) \\ & \text{and } (\epsilon_3, \gamma_3) = (0, 0) \\ (\alpha_i + \beta_i, 0) & \text{if } (\epsilon_1, \gamma_1) = (1, 1) \text{ and } (\epsilon_2, \gamma_2) = (1, 0) \\ & \text{and } (\epsilon_3, \gamma_3) = (0, 1) \\ (\beta_i, \alpha_i) & \text{if } (\epsilon_1, \gamma_1) = (1, 1) \text{ and } (\epsilon_2, \gamma_2) = (0, 1) \\ & \text{and } (\epsilon_3, \gamma_3) = (0, 0) \\ (\alpha_i, \beta_i) & \text{if } (\epsilon_1, \gamma_1) = (0, 1) \text{ and } (\epsilon_2, \gamma_2) = (1, 0) \\ & \text{and } (\epsilon_3, \gamma_3) = (0, 0) \end{cases} \quad (3)$$

Para realizar el conteo de modelos en gráficas tipo árbol se necesita hacer un recorrido primero en profundidad del árbol, es decir, el proceso de conteo de modelos se realiza de las hojas a la raíz, por eso es necesario hacer el recorrido en postorden, asegurando que al llegar a la raíz se han contado los modelos sobre todos los demás vértices. El procedimiento durante el recorrido es como sigue:

Algorithm 1: Algoritmo para conteo de modelos sobre árboles, utilizando los valores asignados a los vértices.

Result: #SAT(F) en v_i

1. Si el vértice v_j es hoja, los valores asignados son $(\alpha_j, \beta_j) = (1, 1)$.
2. Dado el vértice v_j , teniendo un número k de hijos $v_i : 1 \leq i \leq k$, Los modelos se calculan con respecto a las aristas que unen a cada v_i con v_j utilizando la ecuación 1 y se multiplican los valores (α_i, β_i) obtenidos en cada operación.

$$(\alpha_j, \beta_j) = (\prod_{i=1}^k \alpha_i, \prod_{i=1}^k \beta_i) \quad (4)$$

Además de asociar una tupla (α, β) a cada vértice, se asociará un elemento direccional a cada arista de la gráfica.

Gráficas serial paralelo

Las gráficas serial paralelo cumplen con las siguientes características:

- No contienen como subdivisión a la gráfica completa de cuatro vértices K_4 .
- Si es una gráfica simple, debe contener al menos un vértice de grado 2.
- Se construyen a través de dos operaciones:
 - Composición serial: Se divide una arista en la gráfica en dos, incluyendo un vértice nuevo en la gráfica. Dada la arista $e_i = (x, y)$, al dividirse, y agregar el nuevo vértice z , se generan las aristas $e_j = (x, z)$ y $e_k = (z, y)$, sustituyendo la arista original en la gráfica.
 - Composición paralela: Se duplica una arista en la gráfica. Dada la arista $e_i = (x, y)$, se genera una nueva arista $e_j = (x, y)$ y se agrega a la gráfica.

Descomposición de gráficas serial paralelo

Una gráfica serial paralelo se puede deconstruir usando las operaciones de composición de manera inversa como se demuestra en [9] y se resume a continuación.

Descomposición serial

La operación de descomposición serial la definimos como una *contracción* de aristas, en la cual al tener un vértice x_j de grado 2 en la gráfica, $\delta(x_j) = 2$, se toman las dos aristas que lo contienen $e_1 = \{x_i, x_j\}$ y $e_2 = \{x_j, x_k\}$, y se construye la nueva arista $e_3 = \{x_i, x_k\}$. Esta operación permite contraer caminos de n vértices, $P_n = \{\{x_1, x_2\}, \{x_2, x_3\}, \dots, \{x_{n-1}, x_n\}\}$, a una sola arista $\{x_1, x_n\}$.

Descomposición paralela

La operación de descomposición paralela la definimos como una *unión* de aristas, y consiste en reducir un número finito de aristas paralelas uniéndolas en una sola arista, que las represente dentro de la gráfica. Las gráficas serial paralelo

permiten que existan m caminos, entre un mismo par de vértices, y al aplicar la descomposición serial se obtienen un conjunto de m aristas uniendo dos vértices. Al contraer un conjunto de aristas $e_l = \{x_{i_l}, x_{j_l}\}$, $1 \leq l \leq m$, se busca obtener una arista que las representa $\{x_i, x_j\}$.

3. Resultados

En esta sección se muestran las estructuras, métodos y ecuaciones desarrolladas para poder llevar a cabo las operaciones y el conteo de modelos.

Estructuras específicas

A continuación se definen la estructuras necesarias para llevar a cabo el método de conteo.

Elemento direccional

Para una cláusula $\{x_i^\epsilon, x_j^\gamma\}$, una cuádrupla $C_{x_i x_j}$ es asociada a la arista que la representa en la gráfica G . Una cuádrupla $C_{x_i x_j} = \{a_{ij}, b_{ij}, a'_{ij}, b'_{ij}\}$ representa los modelos para las asignaciones posibles de la literal x_j . Inicialmente los valores para esta cuádrupla son tres elementos unitarios y un cero, los cuales representan los tres posibles modelos que una cláusula de dos variables tiene asociada, como se presenta en la ecuación 5:

$$C_{x_i x_j} = \begin{cases} (1, 1, 1, 0) & \text{if } (\epsilon, \gamma) = (1, 1) \\ (1, 0, 1, 1) & \text{if } (\epsilon, \gamma) = (1, 0) \\ (1, 1, 0, 1) & \text{if } (\epsilon, \gamma) = (0, 1) \\ (0, 1, 1, 1) & \text{if } (\epsilon, \gamma) = (0, 0) \end{cases} \quad (5)$$

Esto se puede observar directamente con un ejemplo. Dada la cláusula $(x_1 \vee \bar{x}_2)$, la variable x_1 en disyunción con la negación de la variable x_2 , la tabla de verdad que la representa es:

x_1	x_2	$(x_1 \vee \bar{x}_2)$
0	0	1
0	1	0
1	0	1
1	1	1

Figura 1: Valores de verdad dados para la cláusula $(x_1 \vee \bar{x}_2)$.

Para las posibles combinaciones de signos restantes se puede crear su tabla de manera similar.

Aristas extendidas

El conjunto de aristas extendidas se contruye a partir de las aristas originales de la gráfica, para toda arista $e = \{x_i, x_j\} \in E(G)$ con signos (ϵ, γ) , existe la arista extendida $e' = \{x_i, x_j, C_{x_i x_j}\} \in E'(G)$, donde $C_{x_i x_j}$ se calcula con la ecuación 5.

Vértices extendidos

El conjunto de vértices extendidos se construye con los vértices originales de la gráfica. para todo vértice $x_i \in V(G)$, existe $x_i \in V'(G)$ con una tupla asociada (α_i, β_i) . Los valores de α_i y β_i deben ser inicializados con 1.

Descomposición de gráficas tipo árbol

El objetivo de la descomposición de un árbol es poder reducirlo desde sus hojas hacia la raíz, en este caso se busca poder realizar esta reducción durante la construcción post orden árbol.

Reducción de vértices *hoja*

Dentro de las gráficas tipo árbol o camino, siempre existen vértices de grado 1, y esta operación de *reducción* busca poder eliminar este tipo de vértices de la gráfica. Dado un vértice $x_j \in V(G)$, $\delta(x_j) = 1$, y la arista incidente a el $e_r = \{x_i, x_j\}$, esta operación busca eliminar e_r y x_j de G , $E(G) \setminus e_r$ y $V(G) \setminus x_j$. El objetivo de esta operación es poder llegar a tener un solo elemento en el conjunto de vértices $V(G)$, y que el conjunto $E(G)$ sea vacío.

Conteo sobre una contracción de aristas

Como primera operación para resolver el conteo en las gráficas serial paralelo tenemos la *contracción* de aristas. Esto quiere decir si se tienen dos aristas extendidas, $e'_1 = \{x_i, x_j, C_{x_i x_j}\}$ y $e'_2 = \{x_j, x_k, C_{x_j x_k}\}$, además de un vértice extendido x_j con la tupla asociada (α_j, β_j) . Al realizar la contracción con las aristas e'_1, e'_2 se generará una nueva arista e'_{1-2} uniendo x_i con x_k , $e'_{1-2} = \{x_i, x_k, C_{x_i x_k}\}$, en la cual se calcula el nuevo elemento direccional $C_{x_i x_k}$ con las siguientes ecuaciones:

$$\pi_1(C_{x_i x_k}) = \pi_1(C_{x_j x_k})\pi_1(C_{x_i x_j})\alpha_j + \pi_2(C_{x_j x_k})\pi_3(C_{x_i x_j})\beta_j \quad (6)$$

$$\pi_2(C_{x_i x_k}) = \pi_1(C_{x_j x_k})\pi_2(C_{x_i x_j})\alpha_j + \pi_2(C_{x_j x_k})\pi_4(C_{x_i x_j})\beta_j \quad (7)$$

$$\pi_3(C_{x_i x_k}) = \pi_3(C_{x_j x_k})\pi_1(C_{x_i x_j})\alpha_j + \pi_4(C_{x_j x_k})\pi_3(C_{x_i x_j})\beta_j \quad (8)$$

$$\pi_4(C_{x_i x_k}) = \pi_3(C_{x_j x_k})\pi_2(C_{x_i x_j})\alpha_j + \pi_4(C_{x_j x_k})\pi_4(C_{x_i x_j})\beta_j \quad (9)$$

Donde $\pi_l(C_{x_i x_j})$ es la función de proyección sobre el elemento l de la cuádrupla.

Por ejemplo, dadas las aristas

$e'_1 = \{x_1, x_2, \{1, 0, 1, 1\}\}$ y $e'_2 = \{x_2, x_3, \{1, 1, 1, 0\}\}$ sobre el vértice x_2 con la tupla asociada $(\alpha_2, \beta_2) = (1, 1)$

Generar la arista e'_{1-2} da el elemento direccional $C_{x_1 x_3} = \{2, 1, 1, 0\}$, que como se vió en la sección anterior, $\pi_1(C_{x_1 x_3})$ y $\pi_2(C_{x_1 x_3})$ representan los modelos en los que x_3 aparece positiva, y $\pi_3(C_{x_1 x_3})$ y $\pi_4(C_{x_1 x_3})$ en los que aparece negativa. El total de modelos en la cláusula e'_{1-2} está dado por la suma de los elementos de la cuádrupla $C_{x_1 x_3}$. Tomando la tabla de verdad de estas dos cláusulas:

x_1	x_2	x_3	$(x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Figura 2: Tabla de verdad de la fórmula $(x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_3)$.

Al aplicar el procedimiento de *contracción* en las dos aristas, tenemos como resultado la siguiente tabla:

x_1	x_3	$e_{1-2} = \{(x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_3)\}$
0	0	0
0	1	1
0	0	0
0	1	0
1	0	0
1	1	1
1	0	1
1	1	1

Figura 3: Tabla de verdad de la fórmula $(x_1 \vee \bar{x}_2) \wedge (x_2 \vee x_3)$ eliminando a x_2 de ella.

Se puede observar que la contracción de las cláusulas permite omitir la columna que representa a la variable en común x_2 . Al poder realizar esta *simplificación* de la fórmula, eliminando variables de ella, se reduce su tamaño y por lo tanto también se reduce el tiempo de procesamiento en algoritmos posteriores.

Lemma 1. Sea F una fórmula que representa un camino $P_n = \{\{x_1^{\epsilon_1}, x_2^{\gamma_1}\}, \{x_2^{\epsilon_2}, x_3^{\gamma_2}\} \cdots \{x_{n-1}^{\epsilon_{n-1}}, x_n^{\gamma_{n-1}}\}\}$, dónde la tupla asociada (α_i, β_i) corresponde al elemento x_i , si se aplica la regla de contracción sobre los vértices $x_2, \dots, x_{n-1}$, hasta obtener una tripleta $\{x_1, x_n, C_{x_1 x_n}\}$, entonces:

$$\begin{aligned} \#2SAT(F) = \pi_1(C_{x_1 x_n})\alpha_n + \pi_2(C_{x_1 x_n})\beta_n \\ + \pi_3(C_{x_1 x_n})\alpha_n + \pi_4(C_{x_1 x_n})\beta_n \quad (10) \end{aligned}$$

Demostración. Inicialmente $\alpha_i = \beta_i = 1$, comparándolo con el método de conteo conocido en 1 y de manera inductiva. Teniendo la cláusula simple $\{x_1^{\epsilon}, x_2^{\gamma}\}$ el número de modelos corresponde a los calculados en la ecuación 1 y como se muestra en la figura 4 y en la figura 5 la cuádrupla correspondiente, en ambos casos los modelos calculados son tres.

$$\begin{array}{ccc} (1,1) & & (2,1) \\ x_1^\epsilon & \text{-----} & x_2^\gamma \end{array}$$

Figura 4: Camino P_2 usando la ecuación 1, $\#SAT(F) = \alpha_2 + \beta_2 = 2 + 1 = 3$.

$$\begin{array}{ccc} (1,1) & & (1,1) \\ C_{x_1 x_2}(1,1,1,0) & & \\ x_1 & \text{-----} & x_2 \end{array}$$

Figura 5: Camino P_2 usando conteo sobre contracción de aristas, $\#SAT(F) = \pi_1(C_{x_1 x_2}) + \pi_2(C_{x_1 x_2}) + \pi_3(C_{x_1 x_2}) + \pi_4(C_{x_1 x_2}) = 1 + 1 + 1 + 0$.

En un camino P_3 con aristas $\{\{x_1^{\epsilon_i}, x_2^{\gamma_i}\}, \{x_2^{\epsilon_j}, x_3^{\gamma_j}\}\}$ el número de modelos es 5 si $(\gamma_i = \epsilon_j)$ o 4 si $(\gamma_i \neq \epsilon_j)$.

$$\begin{array}{ccccccc} (1,1) & & (1,1,1,0) & & (1,1) & & (1,1) \\ x_1 & \text{-----} & x_2 & \text{-----} & x_3 & \rightarrow & x_1 & \text{-----} & x_3 \\ & & & & & & (2,1,1,1) & & (1,1) \end{array}$$

Figura 6: Camino P_3 aplicando contracción sobre las aristas de x_2 .

Por hipótesis de inducción $\alpha_n + \beta_n = \pi_1(C_{x_1 x_n}) + \pi_2(C_{x_1 x_n}) + \pi_3(C_{x_1 x_n}) + \pi_4(C_{x_1 x_n})$, por demostrar que $\alpha_{n+1} + \beta_{n+1} = \pi_1(C_{x_1 x_{n+1}}) + \pi_2(C_{x_1 x_{n+1}}) + \pi_3(C_{x_1 x_{n+1}}) + \pi_4(C_{x_1 x_{n+1}})$.

Si $\epsilon = 1$ y $\gamma = 1$, entonces al aplicar la ecuación 1 $\alpha_{n+1} = \alpha_n + \beta_n$ y $\beta_{n+1} = \alpha_n$. Por otro lado $\pi_1(C_{x_1 x_{n+1}}) = 1$, $\pi_2(C_{x_1 x_{n+1}}) = 1$, $\pi_3(C_{x_1 x_{n+1}}) = 1$, $\pi_4(C_{x_1 x_{n+1}}) = 0$, y $\pi_1(C_{x_1 x_n}) + \pi_2(C_{x_1 x_n}) = \alpha_n$, $\pi_3(C_{x_1 x_n}) + \pi_4(C_{x_1 x_n}) = \beta_n$.

Por lo tanto al aplicar las ecuaciones 6, 7, 8 y 9, el valor de

$$\begin{aligned} \pi_1(C_{x_1 x_{n+1}}) &= \pi_1(C_{x_1 x_n}) + \pi_3(C_{x_1 x_n}) = \\ \alpha_n - \pi_2(C_{x_1 x_n}) + \beta_n - \pi_4(C_{x_1 x_n}) &\text{ y para } \\ \pi_2(C_{x_1 x_{n+1}}) &= \pi_2(C_{x_1 x_n}) + \pi_4(C_{x_1 x_n}) = \\ \alpha_n - \pi_1(C_{x_1 x_n}) + \beta_n - \pi_3(C_{x_1 x_n}), &\text{ ahora el valor } \\ \alpha_{n+1} &= \pi_1(C_{x_1 x_{n+1}}) + \pi_2(C_{x_1 x_{n+1}}) = \end{aligned}$$

$$\begin{aligned} \alpha_n - \pi_2(C_{x_1 x_n}) + \beta_n - \pi_4(C_{x_1 x_n}) + \alpha_n - \pi_1(C_{x_1 x_n}) + \beta_n - \pi_3(C_{x_1 x_n}) &= 2\alpha_n + 2\beta_n - \\ (\pi_1(C_{x_1 x_n}) + \pi_2(C_{x_1 x_n})) - (\pi_3(C_{x_1 x_n}) + \pi_4(C_{x_1 x_n})) &= 2\alpha_n + 2\beta_n - \alpha_n - \beta_n = \alpha_n + \beta_n. \end{aligned}$$

Al aplicar las ecuaciones el valor de

$$\begin{aligned} \pi_3(C_{x_1 x_{n+1}}) &= \pi_1(C_{x_1 x_n}) = \alpha_n - \pi_2(C_{x_1 x_n}) \text{ y para } \pi_4(C_{x_1 x_{n+1}}) = \pi_2(C_{x_1 x_n}) = \\ \alpha_n - \pi_1(C_{x_1 x_n}), &\text{ ahora el valor de } \beta_{n+1} = \pi_3(C_{x_1 x_{n+1}}) + \pi_4(C_{x_1 x_{n+1}}) = \alpha_n - \pi_2(C_{x_1 x_n}) + \\ \alpha_n - \pi_1(C_{x_1 x_n}) &= 2\alpha_n - (\pi_1(C_{x_1 x_n}) + \pi_2(C_{x_1 x_n})) = 2\alpha_n - \alpha_n = \alpha_n. \text{ En resumen } \\ \alpha_{n+1} &= \alpha_n + \beta_n \text{ y } \beta_{n+1} = \alpha_n. \end{aligned}$$

El paso inductivo es un análisis sobre los valores de ϵ y γ

Ejemplo, sea $F = \{\{x_1^1, x_2^1\}, \{x_2^1, x_3^0\}, \{x_3^0, x_4^1\}, \{x_4^0, x_5^0\}\}$ una fórmula cuya gráfica restringida es un camino. Las tripletas para cada cláusula son: $\{x_1, x_2, (1, 1, 1, 0)\}$, $\{x_2, x_3, (1, 0, 1, 1)\}$, $\{x_3, x_4, (1, 1, 0, 1)\}$, $\{x_4, x_5, (0, 1, 1, 1)\}$.

La representación que usaremos para las cuádruplas es la siguiente:

$$\begin{array}{ccccccc} (1,1) & & (1,1,1,0) & & (1,1) & & (1,0,1,1) & & (1,1) & & (1,1,0,1) & & (1,1) & & (0,1,1,1) & & (1,1) \\ x_1 & \text{-----} & x_2 & \text{-----} & x_3 & \text{-----} & x_4 & \text{-----} & x_5 \end{array}$$

Primero necesitamos contraer $\{x_1^1, x_2^1\}$ y $\{x_2, x_3^0\}$ en una arista nueva de x_1 a x_3 . Usando las ecuaciones (6-9) obtenemos las nuevas cuádruplas.

$$\begin{array}{ccccccc} (1,1) & & (1,1) & & (1,1) & & (1,1) \\ x_1 & \xrightarrow{(1,1,2,1)} & x_3 & \xrightarrow{(1,1,0,1)} & x_4 & \xrightarrow{(0,1,1,1)} & x_5 \end{array}$$

Ahora se hace la contracción sobre las aristas de x_3 , obteniendo:

$$\begin{array}{ccccccc} (1,1) & & (1,1) & & (1,1) \\ x_1 & \xrightarrow{(3,2,2,1)} & x_4 & \xrightarrow{(0,1,1,1)} & x_5 \end{array}$$

Finalmente con la contracción en las aristas de x_4 se obtiene:

$$\begin{array}{ccccccc} (1,1) & & (1,1) \\ x_1 & \xrightarrow{(2,1,5,3)} & x_5 \end{array}$$

El número de modelos de la fórmula original se obtiene sumando los cuatro elementos de la cuádrupla, multiplicando por (α_5, β_5) :

$\#2SAT(F) = 2 * 1 + 1 * 1 + 5 * 1 + 3 * 1$, dando como resultado 11 modelos.

Conteo en una unión de aristas

Si $e_1, \dots, e_h$ son las aristas paralelas sobre el mismo par de vértices x_i, x_j , $e'_l = \{x_i, x_j, C_{x_i x_j}\} : 1 \leq l \leq h$. La unión puede llevarse a cabo con la ecuación:

$$\pi_k(C_{x_i x_j}) = \prod_{l=1}^h \pi_k(C_{x_i x_j}) \quad (11)$$

Donde $k = 1, 2, 3, 4$, y se puede formar la nueva arista $e_{1-h} = \{x_i, x_j, C_{x_i x_j}\}$

Lemma 2. Sea F una fórmula representada por una gráfica con dos o más cláusulas paralelas e.g. $F = \{\{x_1^{\epsilon_1}, x_2^{\gamma_1}\}, \{x_1^{\epsilon_2}, x_2^{\gamma_2}\}\}$ o la fórmula $F = \{\{x_1^{\epsilon_1}, x_2^{\gamma_1}\}, \{x_1^{\epsilon_2}, x_2^{\gamma_2}\}, \dots, \{x_1^{\epsilon_m}, x_2^{\gamma_m}\}\}$, se aplica la unión de aristas hasta obtener la arista que las represente a todas $(x_1, x_2, C_{x_1 x_2})$ entonces:

$$\begin{aligned} \#2SAT(F) = \pi_1(C_{x_1 x_2})\alpha_2 + \pi_2(C_{x_1 x_2})\beta_2 \\ + \pi_3(C_{x_1 x_2})\alpha_2 + \pi_4(C_{x_1 x_2})\beta_2 \end{aligned} \quad (12)$$

Demostración. Un caso de análisis simple comparando los resultados de los algoritmos conocidos de las ecuaciones 2 y 3. El resultado para dos cláusulas paralelas se muestra en la figura 7 (mostrando sólo el elemento direccional $C_{x_i x_j}$) y su unión en la figura 8.

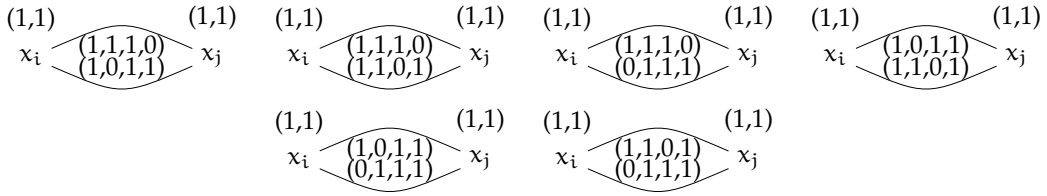


Figura 7: Casos paralelos sobre dos cláusulas.

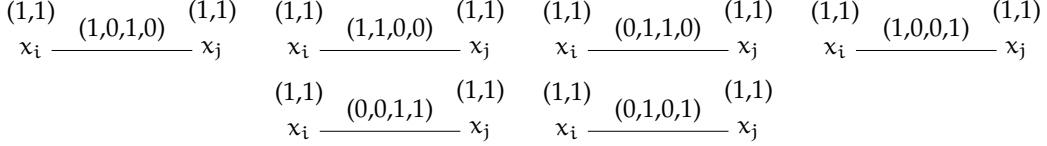


Figura 8: Aplicando conteo sobre unión de aristas en los casos de dos cláusulas.

De la ecuación 2 el par (α_i, β_i) obtenido en casos de dos cláusulas paralelas $(x_i^{\epsilon_1}, x_j^{\gamma_1})$ y $(x_i^{\epsilon_2}, x_j^{\gamma_2})$ son $(2, 0)$ si $(\epsilon_1, \gamma_1) = (1, 1)$ y $(\epsilon_2, \gamma_2) = (0, 1)$, Es $(0, 2)$ si $(\epsilon_1, \gamma_1) = (1, 0)$ y $(\epsilon_2, \gamma_2) = (0, 0)$, y $(1, 1)$ en los demás casos. Para el análisis sobre tres cláusulas es similar.

Conteo sobre una reducción de vértice

Para realizar el conteo de modelos en gráficas tipo árbol se necesita hacer un recorrido post orden del árbol (no necesariamente binario), es decir, el proceso de conteo de modelos se realiza de las hojas a la raíz. El procedimiento durante el recorrido es como sigue:

Algorithm 2: Algoritmo para conteo de modelos sobre árboles, utilizando aristas y vértices extendidos.

Result: #SAT(F) en x_i

1. Si el vértice asociado es hoja los valores asociados al vértice son $(\alpha_j, \beta_j) = (1, 1)$;
2. Dado el vértice x_i , teniendo un número k de hijos $x_j : 1 \leq j \leq k$, y el conjunto de aristas con elemento direccional $C_{x_j x_i}$, los valores (α_i, β_i) se calculan usando las ecuaciones 13 y 14;

$$\alpha_i = \prod_{j=1}^k (\pi_1(C_{x_j x_i}) \alpha_j + \pi_2(C_{x_j x_i}) \beta_j) \quad (13)$$

$$\beta_i = \prod_{j=1}^k (\pi_3(C_{x_j x_i}) \alpha_j + \pi_4(C_{x_j x_i}) \beta_j) \quad (14)$$

Lemma 3. Sea F una fórmula representada por una gráfica tipo árbol, con raíz x_i , entonces: #SAT(F) se calcula con el algoritmo 2

Demostración. Ya que el algoritmo 1 es correcto, la prueba consiste en comparar que la salida del algoritmo 2 es equivalente a la salida del algoritmo 1.

Ya que los signos en la recurrencia 1 para el algoritmo 1, se reemplazan por las cuádruplas en el algoritmo 2, entonces las operaciones son equivalentes.

Ejemplo, un caso de análisis sobre una gráfica tipo árbol con cinco vértices usando la ecuación 4, y las ecuaciones 13 y 14 se muestran en las figuras 9 y 10.

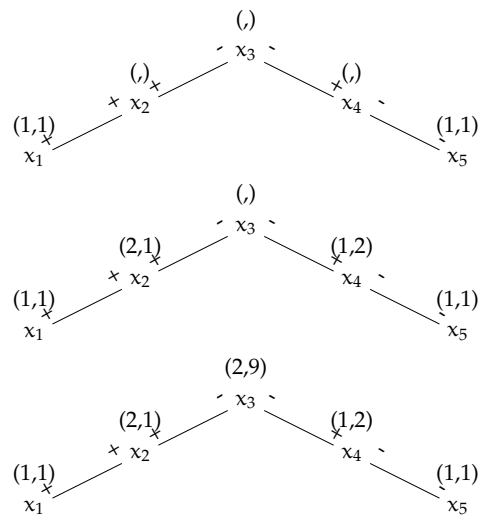


Figura 9: Representación y conteo sobre una gráfica tipo árbol con cinco vértices, tomando como raíz x_3 , usando la ecuación 4.

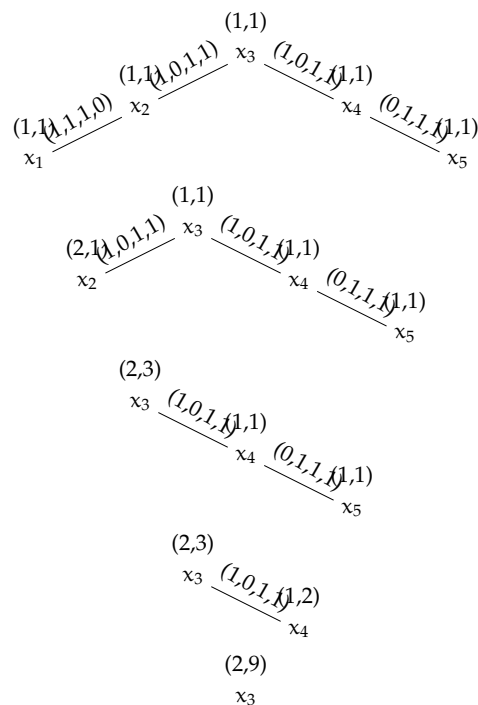


Figura 10: Representación y conteo sobre una gráfica tipo árbol con cinco vértices, tomando como raíz x_3 , usando las ecuaciones 13 y 14.

Conteo sobre árboles con subgráficas serial paralelo

El método para realizar el conteo sobre árboles con subgráficas tipo serial paralelo, sigue los siguientes pasos:

- Identificar el árbol de la gráfica, para saber cuál será el vértice raíz.
- Identificar los puntos *origen* y *sumidero* de las subgráficas serial paralelo dentro del árbol.
- Reducir las subgráficas serial paralelo con los métodos presentados en las secciones 3 y 3.
- Recorrer el árbol y aplicar el algoritmo 2.

Theorem 1. Si F es una fórmula tipo árbol con subfórmulas tipo serial paralelo, entonces el método de conteo calcula $\#2SAT(F)$ de manera correcta.

Demostración. Por los lemmas 1 y 2 se demuestra el conteo sobre gráficas serial paralelo y por el lemma 3 el conteo sobre gráficas tipo árbol.

Ejemplo

Ahora usaremos un ejemplo para aplicar nuestro algoritmo.

Dada la fórmula $F = \{\{x_1^1, x_3^1\}, \{x_{10}^1, x_1^1\}, \{x_2^1, x_3^1\}, \{x_{10}^1, x_2^1\}, \{x_4^1, x_3^1\}, \{x_3^1, x_8^1\}, \{x_{10}^1, x_4^1\}, \{x_9^1, x_{10}^1\}, \{x_7^1, x_{10}^1\}, \{x_6^1, x_5^1\}, \{x_5^1, x_{10}^1\}, \{x_{12}^1, x_{13}^1\}, \{x_{12}^1, x_{14}^1\}, \{x_{12}^1, x_{11}^1\}, \{x_{11}^1, x_8^1\}, \{x_{14}^1, x_8^1\}, \{x_{13}^1, x_8^1\}\}$

Usaremos la representación gráfica de la fórmula usando vértices extendidos y el conjunto de aristas extendidas $E' = \{\{x_1, x_3, (1, 1, 1, 0)\}, \{x_{10}, x_1, (1, 1, 1, 0)\}, \{x_2, x_3, (1, 1, 1, 0)\}, \{x_{10}, x_2, (1, 1, 1, 0)\}, \{x_4, x_3, (1, 1, 1, 0)\}, \{x_3, x_8, (1, 1, 1, 0)\}, \{x_{10}, x_4, (1, 1, 1, 0)\}, \{x_9, x_{10}, (1, 1, 1, 0)\}, \{x_7, x_{10}, (1, 1, 1, 0)\}, \{x_6, x_5, (1, 1, 1, 0)\}, \{x_5, x_{10}, (1, 1, 1, 0)\}, \{x_{12}, x_{13}, (1, 1, 1, 0)\}, \{x_{12}, x_{14}, (1, 1, 1, 0)\}, \{x_{12}, x_{11}, (1, 1, 1, 0)\}, \{x_{11}, x_8, (1, 1, 1, 0)\}, \{x_{14}, x_8, (1, 1, 1, 0)\}, \{x_{13}, x_8, (1, 1, 1, 0)\}\}$:

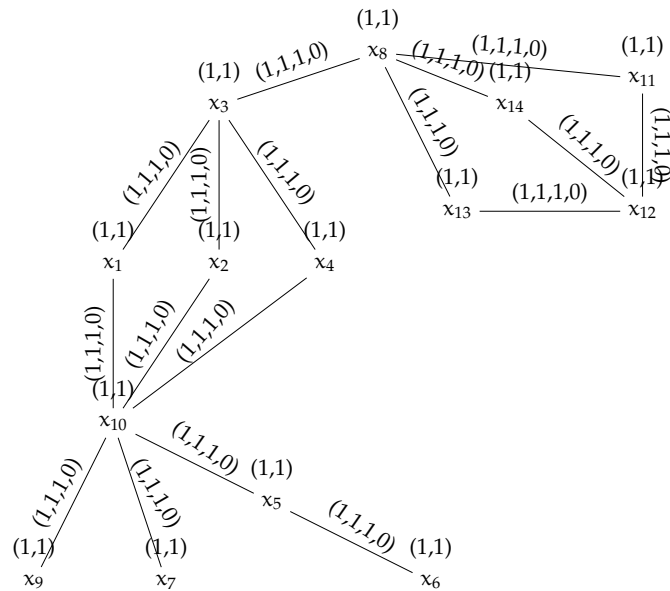


Figura 11: Representación gráfica de F.

Tomando la representación gráfica de la figura 11, tomaremos como raíz de la gráfica a x_8 y tenemos dos subgráficas serial paralelo, la primera tiene como vértices *origen* y *sumidero* a los vértices x_{10} y x_3 respectivamente, y la segunda tiene como vértices *origen* y *sumidero* a x_{12} y x_8 .

El proceso de reducción de estas subgráficas serial paralelo se muestra en los lemas 1 y 2, y el resultado se ilustra en la figura 12.

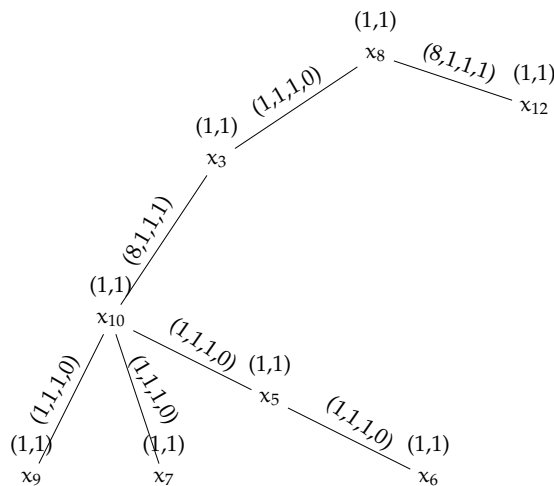


Figura 12: Representación gráfica de F al aplicar la reducción de las subgráficas serial paralelo.

Posteriormente se realiza el recorrido del árbol aplicando el conteo por reducción de vértice, como se muestra en las siguientes figuras.

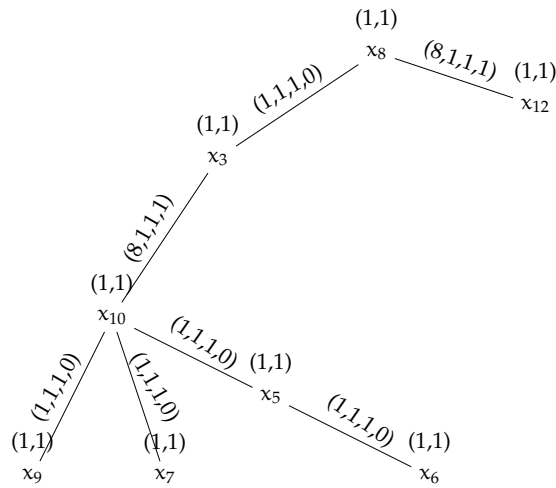


Figura 13: Árbol sobre el que se aplicará el procedimiento de conteo.

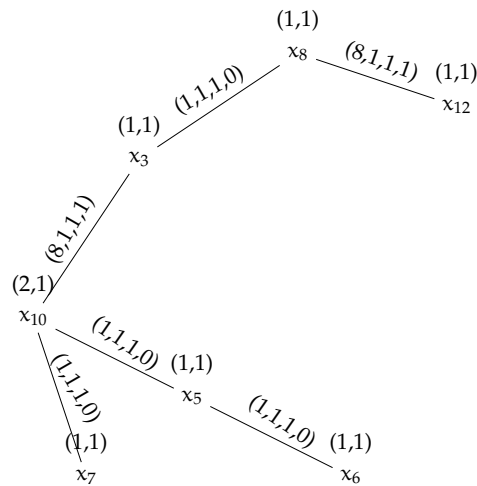


Figura 14: Resultado de la reducción del vértice x_9 .

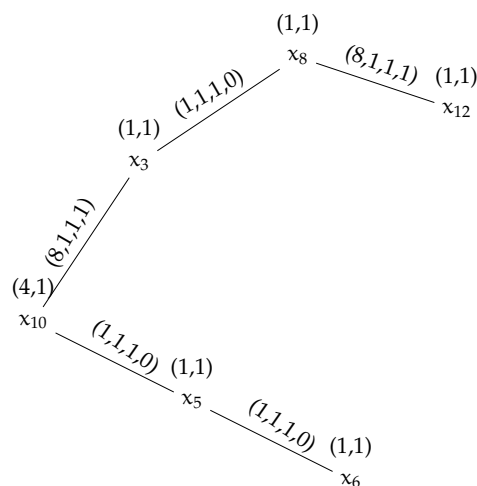


Figura 15: Resultado de la reducción del vértice x_7 .

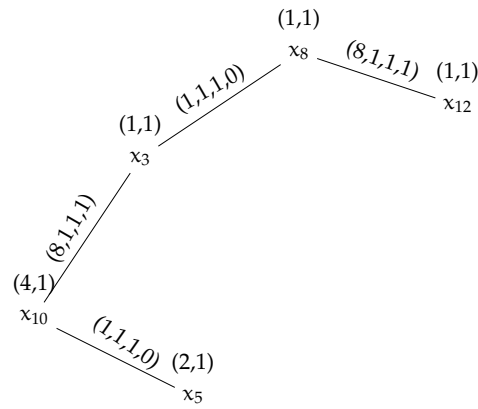


Figura 16: Resultado de la reducción del vértice x_6 .

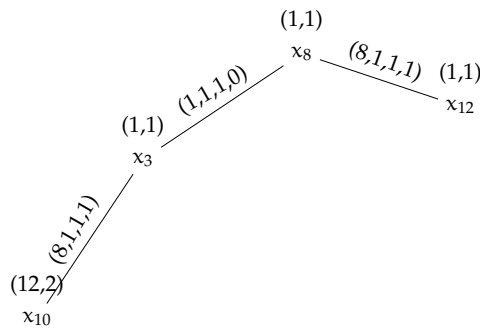


Figura 17: Resultado de la reducción del vértice x_5 .

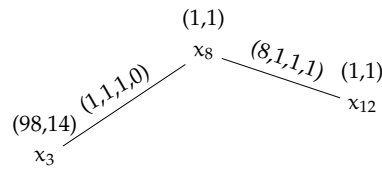


Figura 18: Resultado de la reducción del vértice x_{10} .

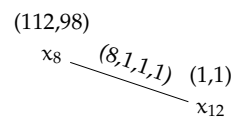


Figura 19: Resultado de la reducción del vértice x_3 .

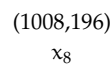


Figura 20: Resultado de la reducción del vértice x_{12} .

Como resultado del ejemplo se obtiene un total de 1204 modelos en la fórmula F.

Aunque ya existían métodos para contar sobre árboles como los presentados en [8], este método permite conservar los modelos que representan subgráficas tipo serial paralelo, en una sola arista, con lo que se puede aplicar la operación de

reducción por vértice y obtener el conteo de modelos correspondiente. Con esto se logra extender el tipo de fórmulas sobre las que se puede contar modelos en tiempo lineal.

4. Conclusiones

En este artículo se presento una nueva forma de contar modelos sobre gráficas tipo camino y árbol. El nuevo método permite contar sobre gráficas tipo árbol que contiene subgráficas serial-paralelo dentro de él. Adicionalmente, el procedimiento presentado se puede realizar en tiempo lineal, ya que como demostraron en [9] contar sobre gráficas serial paralelo se puede hacer en tiempo lineal debido a que el procedimiento utiliza el número de aristas y vértices en la gráfica serial paralelo ($O(n + m)$), y el conteo sobre gráficas tipo árbol también es lineal ya que se hace un recorrido en post orden sobre sus vértices ($O(n)$).

Bibliografía

- [1] P. Winkler G. Brifhtwell. Counting linear extensions. *Order*, 8(e):225–242, 1991.
- [2] M. A. López-Medina, J. R. Marcial-Romero, G. De Ita Luna, H. A. Montes-Venegas, and R. Alejo. A linear time algorithm for solving #2SAT on cactus formulas. *CoRR, ams/1702.08581*, 2017.
- [3] M. A. López-Medina, J. R. Marcial-Romero, G. De Ita, and Y. Moyao. A Linear Time Algorithm for Computing #2SAT for Outerplanar 2-CNF Formulas. *Lecture Notes in Computer Science*, 10880:72–81, 2018.
- [4] M. Wahlström. A tighter bound for counting max-weight solutions to 2sat instances. *Springer Berlin Heidelberg*, pages 202–213, 2008.
- [5] B. Schoenmakers and L. A.M. A new algorithm for the recognition of series parallel graphs. *CWI (Centre for Mathematics and Computer Science)*, 1995.
- [6] D. Eppstein. Parallel recognition of series-parallel graphs. *Information and Computation*, 98:41 – 55, 1992.
- [7] K. Takamizawa, T. Nishizeki, and N. Saito. Linear-time computability of combinatorial problems on series-parallel graphs. *Journal of the Association for Computing Machinery*, 29(3):623 – 641, 1982.
- [8] J. R. Marcial-Romero, G. De Ita, J. A. Hernández, and R. M. Valdovinos. A parametric polynomial deterministic algorithm for #2sat. *Lecture Notes in Computer Science*, 9413:202–213, 2015.
- [9] Marco A. López Medina, José Raymundo Marcial-Romero, Guillermo De Ita Luna, and José A. Hernández. A linear time algorithm for counting #2sat on series-parallel formulas. 12468:437–447, 2020.

Direcciones de correo de los autores

LÓPEZ MEDINA, M. A.: malopezme@uaemex.mx

MARCIAL ROMERO, J. R.: jrmarcialr@uaemex.mx

GONZÁLEZ RUIZ, J. L.: jlgonzalezru@uaemex.mx

HERNÁNDEZ SERVÍN, J. A.: xoseahernandez@uaemex.mx