

Conteo de modelos en la clase sintáctica 2μ -3MON

Carlos Guillén, Rafael Lemuz, e Irene Ayaquica

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Puebla,
México

{cguillen,rlemuz,ayaquica}@cs.buap.mx

Resumen. El problema de conteo de modelos en fórmulas Booleanas es un problema $\#P$ -completo, es decir, no se conocen algoritmos deterministas en el modelo clásico de computabilidad (máquinas de Turing) que realice este conteo con complejidad en tiempo polinomial. La dificultad persiste aún imponiendo condiciones más restrictivas sobre las clases sintácticas de fórmulas Booleanas. En este artículo presentamos una familia tratable dentro de la clase sintáctica 2μ -3MON. La identificación de esta familia se hace a través del hipergrafo asociado a estructuras simples como cadenas y ciclos. Se identifican también operadores matriciales que actúan sobre estas estructuras; estos operadores conducen a algoritmos eficientes que efectúan el conteo de modelos sobre la familia identificada en tiempo lineal con respecto al número de cláusulas de la fórmula instanciada, a diferencia de los métodos basados en invariantes hipergráficos (como el ancho de árbol) que realizan este conteo en tiempo cúbico.

Palabras clave. $\#SAT$, clase sintáctica, hipergrafo.

Model Counting in the 2μ -3MON Syntactic Class

Abstract. The counting model problem in Boolean formulas is $\#P$ -complete, i.e., there is no known deterministic algorithm in the classical computability model (Turing machine) that makes this count in polynomial time. The difficulty persists even imposing more restrictive conditions on the syntactic classes of Boolean formulas. In this paper we present a treatable family within the syntactical class 2μ -3MON. The identification of this family is done by using the hypergraph associated with simple structures such as chains and cycles. Then, matrix operators acting over these structures are identified; these operators lead to efficient algorithms that perform the model counting on the identified family in linear time for the number of clauses in the instantiated formula; unlike hypergraphic invariant based methods (such as tree width), which perform the count in cubic time.

Keywords. $\#SAT$, syntactic class, hypergraph.

1. Introducción

El problema de computar el número de modelos de una fórmula Booleana es conocido como el problema $\#SAT$ y surge como una extensión natural del problema de decisión de satisfactibilidad (problema SAT), que fue introducido en el campo de la complejidad computacional en 1971 por Cook [4]. El problema $\#SAT$ además de ser un problema interesante por sí mismo en el ámbito de la teoría de la computación, resulta también interesante por su aplicación a diversos problemas en otras disciplinas. Por mencionar algunos ejemplos, en inteligencia artificial la actualización de una base de conocimiento expresada en términos de fórmulas Booleanas puede realizarse a través del cálculo de revisión de creencias que efectúa un agente inteligente cuando surge nueva información que se contrapone con parte de la información de la base [11]. En estructuras moleculares, resulta interesante determinar el índice de Merrifield-Simmons [10], este concepto equivale a contar conjuntos independientes en un grafo o determinar el número de modelos en una fórmula Booleana monótona formada a través de conjunciones de cláusulas con dos literales [16].

En su generalidad, $\#SAT$ es un problema difícil. Valiant [17] demuestra que este problema se encuentra ubicado en la clase $\#P$ -completo, aún más, se ha demostrado que este problema es difícil de aproximar; es decir, no se conocen algoritmos de aproximación aleatoria que se aproximen al número de modelos de una fórmula Booleana dentro de cierto radio ρ , dependiente del tamaño de la instancia del problema, con probabilidad $\geq 3/4$ [14]. Debido a esto existen diferentes formas de restringir o tratar este problema [5, 6, 9].

Un primer caso de simplificación es considerar fórmulas Booleanas en forma normal conjuntiva (CNF) como fórmulas de entrada al problema. Si \mathcal{C} denota una subclase sintáctica de fórmulas

Booleanas en CNF, entonces con $\#\mathcal{C}$ denotamos su correspondiente problema de conteo, es decir, el problema de contar modelos de fórmulas pertenecientes a la clase \mathcal{C} . Un segundo caso es el de imponer restricciones sobre el número de literales o el número de apariciones de variables en las cláusulas o fórmulas, respectivamente. Por ejemplo, si cada cláusula de Φ tiene a lo más k literales, Φ se considera una fórmula en la clase k SAT, k MON para fórmulas en donde cada una de sus cláusulas tiene a lo más k literales y ninguna literal aparece negada y k HORN para fórmulas con cláusulas HORN de k literales; una cláusula es HORN si a lo más tiene una literal no negada. Si cada variable en la fórmula Φ aparece a lo más k veces, entonces Φ se considera en la clase $k\mu$ -SAT. Con $l\mu$ - k MON (HORN) denotamos la clase sintáctica de fórmulas monótonas (HORN) donde cada variable aparece a lo más l veces y cuyas cláusulas se constituyen de a lo más k literales.

Por otra parte, los algoritmos para resolver el problema $\#$ SAT se pueden clasificar en exactos o aproximados—un algoritmo es exacto si para toda instancia de entrada siempre se encuentra la solución buscada mientras que, un algoritmo es aproximado si la solución que se halla está dentro de un intervalo cercano al valor o valores que se buscan—. Los resultados de los algoritmos exactos, se pueden agrupar en dos clases, una que contiene los algoritmos exactos de tiempo polinomial y otra en donde sólo se conocen algoritmos de tiempo exponencial.

Existen pocos problemas en la jerarquía de restricciones sintácticas, para los cuales se conoce que el conteo exacto del número de modelos para una fórmula Booleana se puede realizar en tiempo polinomial, de hecho, se tienen los siguientes resultados relevantes. El primero se debe a Vadhan [16], quien demostró que $\#2\mu$ -2MON puede ser resuelto usando fórmulas de recurrencia. El segundo se debe a Roth [13], quien generaliza este resultado a $\#2\mu$ -2SAT. Sus algoritmos utilizan técnicas combinatorias que se han mejorado considerando el problema como un problema de orientación de grafos libres de sumideros (un sumidero es un vértice con grado externo cero) [2]. Por su parte en Fischer et. al [8] se muestra que si una fórmula tiene ancho de árbol acotado o ancho de clique acotado entonces se puede establecer en tiempo polinomial el número de modelos. Un resultado similar se presenta

en [12], basado en la detección de conjuntos *backdoor* fuertes de tamaño limitado.

En el estudio de las clases sintácticas, existen muchos resultados obtenidos para $\#$ SAT que están en la clase de complejidad $\#$ P-completo. Comenzando por los resultados de Valiant [17], donde se demuestra que dada una fórmula Booleana Φ , el contar el número de asignaciones que la satisfacen es un problema $\#$ P-completo. Roth en [13] demuestra usando una técnica simple de reescritura que uno puede, a través de una reducción parsimoniosa de $\#2$ MON a $\#4\mu$ -2HORN, para llegar a que éste último es un problema $\#$ P-completo. Una función $f : \Sigma_1^* \rightarrow \mathbb{N}$ tiene una reducción parsimoniosa con $g : \Sigma_2^* \rightarrow \mathbb{N}$ si existe una función $\sigma : \Sigma_1^* \rightarrow \Sigma_2^*$ computable en tiempo polinomial tal que $f = \sigma \circ g$, una cláusula es de Horn si como máximo tiene una literal positiva. Vadhan en [15] demuestra que $\#4\mu$ -2MON es $\#$ P-completo; más tarde, fortaleció este mismo resultado en el escenario de conjuntos independientes (conjuntos de vértices en un grafo que no contienen ambos puntos extremos de cualquier arista) para demostrar que también el conteo de conjuntos independientes en grafos bipartitos planos de grado máximo 4 es $\#$ P-completo [16]. Dyer y Greenhill en [7] presentan resultados que mejoran los obtenidos por Vadhan mostrando que $\#3\mu$ -2MON es $\#$ P-completo. Finalmente, Buble y Dyer demuestran en [3] que $\#2\mu$ -SAT es $\#$ P-completo y establecen que $\#2\mu$ -MON también es $\#$ P-completo. Dentro de las versiones de $\#$ SAT cuya clase de complejidad aún permanece sin determinarse, están $\#2\mu$ -3MON y $\#2\mu$ - k SAT con $k > 2$.

En el presente artículo se propone, utilizando la idea de conteo de modelos mediante operadores matriciales, una familia de funciones tratable dentro de esta clasificación sintáctica. La tratabilidad de esta familia queda determinada de acuerdo a la estructura topológica del hipergrafo asociado a la fórmula. Un hipergrafo es un par de conjuntos disjuntos $G = (V, E)$, donde V es llamado conjunto de vértices o nodos y E , llamado conjunto de hiperaristas, tiene como elementos subconjuntos de V que pueden tener cardinalidad arbitraria, si todos las hiperaristas tienen cardinalidad 2, G se llama grafo y E conjunto de aristas. Esta estructura de hipergrafo es adecuada para fórmulas en CNF con cláusulas

de más de dos variables, ya que éstas son identificadas con las hiperaristas de un hipergrafo.

La contribución principal de este trabajo es la identificación, mediante operadores matriciales, de subclases sintácticas tratables correspondientes a hipergrafos que se presentan como cadenas y ciclos; aquí las cadenas son secuencias finitas de hiperaristas, donde dos hiperaristas consecutivas tienen uno o dos nodos comunes y son ajenas si no son consecutivas, la cadena es simple o alternada dependiendo si sus hiperaristas consecutivas coinciden en un nodo o si se combinan con hiperaristas consecutivas coincidentes en dos nodos. Un ciclo es la unión de una cadena con una hiperarista que conecta los extremos de ésta, también el ciclo puede ser simple o alternado en dependencia de si sus hiperaristas consecutivas se conectan en uno o dos nodos.

La organización de este documento fue planeada de la siguiente manera. La sección I, se reserva para la introducción y motivación del problema; en la sección II, se introducen la notación y los conceptos básicos del trabajo. En las secciones III y IV, se presentan los operadores que permiten realizar el conteo de modelos sobre las estructuras identificadas. En la sección V, se presenta una discusión sobre la complejidad de los algoritmos que se pueden derivar de nuestros resultados; así como la relación que existe con los ancho-conceptos. Finalmente, en la sección VI, se comentan las conclusiones y el trabajo futuro.

2. Modelos y cargas

En esta sección se introducen los conceptos de *carga* y *carga conjunta* de una y dos variables respectivamente, relativas a una fórmula Booleana. Se establecen algunos resultados simples sobre el conjunto de modelos de una fórmula Booleana que serán de utilidad en las secciones que siguen. De aquí en adelante cuando se mencione la palabra “fórmula”, se deberá entender “fórmula Booleana”.

El conjunto de variables de una fórmula Φ se representa mediante la notación $Var(\Phi)$ y $Lit(\Phi)$ denota su conjunto de literales, una cláusula c con literales ℓ_1, \dots, ℓ_k . se denota como $c = \{\ell_1, \dots, \ell_k\}$, una fórmula Φ en CNF con cláusulas c_1, \dots, c_m se escribe como $\Phi = \{c_1, \dots, c_m\}$. La fórmula Φ puede ser identificada con un hipergrafo G_Φ cuyo

conjunto de vértices es $Lit(\Phi)$ ($Var(\Phi)$ para el caso monótono) y con hiperaristas en el conjunto $\{c_1, \dots, c_m\}$, en consecuencia, si se menciona alguna propiedad gráfica sobre la fórmula Φ , se deberá entender que dicha propiedad se refiere al hipergrafo asociado G_Φ , por ejemplo, que la fórmula Φ sea conectada significa que el hipergrafo G_Φ es conectado, esto es, para cualesquiera dos nodos a y b distintos de G_Φ existe una secuencia finita de hiperaristas c_1, \dots, c_k en G_Φ tales que $a \in c_1, b \in c_k, c_i \cap c_{i+1} \neq \emptyset$ y $c_i \cap c_j = \emptyset$ si $|i - j| > 1$. Si A es un conjunto, $|A|$ denota su cardinalidad, $M(\Phi)$ es el conjunto de modelos de la fórmula Φ y $M_{C(x_1, x_2, \dots, x_n)}(\Phi)$ es el conjunto de modelos con la restricción C sobre las variables x_1, x_2, \dots, x_n de Φ . Por ejemplo, dada una fórmula Φ , $M_{x=1}(\Phi)$ es el conjunto de modelos de Φ tales que tienen el valor de 1 en la variable x y $M_{x=1, y=0}(\Phi)$ es el conjunto de modelos de Φ tales que toman el valor de 1 en la variable x y el valor de 0 en la variable y . Observe que $M_{C(x_1, x_2, \dots, x_n)}(\Phi) \subseteq M(\Phi)$, para cualquier restricción C , esto es $C(x_1, x_2, \dots, x_n) = \{x_1 = i_1, x_2 = i_2, \dots, x_n = i_n\}, i_k \in \{0, 1\}$. En lo que sigue \mathbb{N} denota el conjunto de los números naturales y \mathbb{N}^2 son parejas ordenadas (m, n) de naturales que serán denotadas indistintamente como vectores renglón (m, n) o vectores columna $\begin{pmatrix} m \\ n \end{pmatrix}$. Si q denota una matriz o vector, será denotado con negrita q .

Dada una fórmula Φ , con $\Phi_{C(x_1, x_2, \dots, x_n)}$ denotamos la fórmula que resulta de simplificar Φ al asignar parcialmente los valores de la restricción a las variables x_1, x_2, \dots, x_n . Por ejemplo, para $i \in \{0, 1\}$, $\Phi_{x=i}$ denota la fórmula que resulta de simplificar Φ al asignar parcialmente el valor de i a la variable x . En el caso particular de $\Phi = \{\{x, y\}, \{y, w, z\}, \{x, u\}\}$, otros ejemplos son: $\Phi_{x=0} = \{\{y\}, \{y, w, z\}, \{u\}\}$, $\Phi_{x=1} = \{\{y, w, z\}\}$ y $\Phi_{x=0, z=1} = \{\{y\}\}$.

Sea $R_{x=i}(\Phi)$ el conjunto de variables diferentes de x eliminadas cuando Φ es simplificada a $\Phi_{x=i}$, esto es,

$$R_{x=i}(\Phi) = Var(\Phi) \setminus (\{x\} \cup Var(\Phi_{x=i})).$$

En el ejemplo precedente, $R_{x=1}(\Phi) = \{u\}$. Para dos fórmulas Φ y Φ' tales que la intersección de los conjuntos $Var(\Phi')$ y $Var(\Phi)$ contiene únicamente a $\{x\}$ o es vacía, se cumplen:

$$Var((\Phi \cup \Phi')_{x=i}) = Var(\Phi_{x=i}) \cup Var(\Phi'_{x=i}) \text{ y } (1)$$

$$Var(\Phi_{x=i}) \cap Var(\Phi'_{x=i}) = \emptyset. \quad (2)$$

Dada una fórmula Φ e $i \in \{0, 1\}$, tenemos

$$|M_{x=i}(\Phi)| = 2^{|R_{x=i}(\Phi)|} |M(\Phi_{x=i})| \quad (3)$$

DEFINICIÓN 2.1. Dada una fórmula Φ y una variable $x \in Var(\Phi)$, la *carga* de x relativa a Φ , es la pareja ordenada (m, n) , tal que $m = |M_{x=1}(\Phi)|$ y $n = |M_{x=0}(\Phi)|$, esta pareja es denotada como $\#sat(\Phi, x)$.

La siguiente definición es de utilidad cuando se tiene información sobre el número de modelos de una fórmula concentrada en dos variables.

DEFINICIÓN 2.2. Dada una fórmula Φ y dos variables $x, y \in Var(\Phi)$, la *carga conjunta* de x relativa a Φ , es la matriz $(m_{i,j})$, tal que $m_{i,j} = |M_{x=i,y=j}(\Phi)|$, $i, j \in \{0, 1\}$, esta matriz es denotada como $\#sat(\Phi, x, y)$.

Observación 1. En general, si denotamos con A_i el conjunto de valores en $\{0, 1\}^n$ donde se cumple la restricción $C_i(x_1, x_2, \dots, x_n)$ para $i \in \{0, 1\}$, y si A_1 y A_2 son disjuntos, entonces $M_{C_1 \cup C_2}(\Phi) = M_{C_1}(\Phi) \cup M_{C_2}(\Phi)$, de donde se sigue que

$$|M_{C_1 \cup C_2}(\Phi)| = |M_{C_1}(\Phi)| + |M_{C_2}(\Phi)|.$$

Dadas x, y variables de Φ , las siguientes igualdades son un ejemplo de la observación 1.

- 1.- $|M(\Phi)| = |M_{x=1}(\Phi)| + |M_{x=0}(\Phi)|$
- 2.- $|M(\Phi)| = |M_{x=1,y=1}(\Phi)| + |M_{x=1,y=0}(\Phi)| + |M_{x=0,y=1}(\Phi)| + |M_{x=0,y=0}(\Phi)|$
- 3.- $|M_{x=1}(\Phi)| = |M_{x=1,y=1}(\Phi)| + |M_{x=1,y=0}(\Phi)|$
- 4.- $|M_{x=0}(\Phi)| = |M_{x=0,y=1}(\Phi)| + |M_{x=0,y=0}(\Phi)|$

y a su vez de estas igualdades tenemos las siguientes relaciones matriciales.

$$\#sat(\Phi, x) = \#sat(\Phi, x, y) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (4)$$

Por lo tanto, el problema $\#SAT$ se puede ver como el problema de hallar la carga de una variable de la fórmula instanciada. Note también que si se pueden obtener algunas cargas de las variables de una fórmula Φ se tiene información más precisa que la que proporciona el número $|M(\Phi)|$, esto es, no sólo se conoce el número de modelos de la fórmula, sino también se sabe el número de modelos donde una determinada variable toma el valor 1 y el número de modelos donde esta misma variable toma el valor 0. Esta información, como

se verá más adelante, ayuda a obtener algoritmos más directos y pruebas más concisas sobre los resultados presentados en este artículo.

La siguiente definición presenta una descripción de una operación que se encuentra directamente relacionada con el conteo de modelos. Esta operación tiene la característica de relacionar cargas de variables y de permitir ciertas reducciones en una fórmula dada.

DEFINICIÓN 2.3. Sea $\mathbf{A} = (a_{ij})$ y $\mathbf{B} = (b_{ij})$ matrices de $n \times m$ con entradas en \mathbb{N} , el producto de Hadamard de la matriz \mathbf{A} con la matriz \mathbf{B} se define como la matriz

$$\mathbf{A} \odot \mathbf{B} = (a_{ij}b_{ij}) \quad (5)$$

Se verifica fácilmente que el producto “ \odot ” es asociativo, distributivo y conmutativo.

Una componente de una fórmula Φ es una subfórmula Φ' conectada maximal. Si Φ_1 y Φ_2 son distintas componentes de Φ , es claro que

$$|M(\Phi)| = |M(\Phi_1)| |M(\Phi_2)| \quad (6)$$

Entonces para simplificar, se puede suponer que toda fórmula es conectada. La complejidad en tiempo de un procedimiento para una fórmula arbitraria no es afectada, ya que los procedimientos para determinar las componentes pueden llevarse a cabo en tiempo lineal [2].

El siguiente lema establece que si conocemos las cargas de una variable con respecto a dos fórmulas que tienen sólo a esta variable en común, entonces la carga de la unión de dichas fórmulas puede ser calculada mediante el producto de Hadamard de las cargas respectivas (ver figura 1).

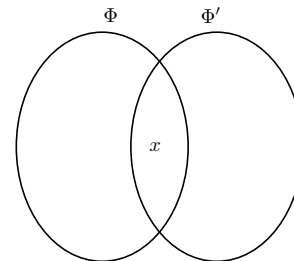


Fig. 1. Fórmulas que coinciden en la variable x

LEMA 2.4. Si $Var(\Phi) \cap Var(\Phi') = \{x\}$ entonces

$$\#sat(\Phi \cup \Phi', x) = \#sat(\Phi, x) \odot \#sat(\Phi', x).$$

DEMOSTRACIÓN. Por definición $\#\text{sat}(\Phi \cup \Phi', x) =$

$$(|M_{x=1}(\Phi \cup \Phi')|, |M_{x=0}(\Phi \cup \Phi')|),$$

como $\text{Var}(\Phi_{x=i}) \cap \text{Var}(\Phi'_{x=i}) = \emptyset$ para cada $i \in \{0, 1\}$, de la ecuación (3) y se sigue la cadena de igualdades:

$$\begin{aligned} |M_{x=i}(\Phi \cup \Phi')| &= \\ &= 2^{|R_{x=i}(\Phi \cup \Phi')|} |M((\Phi \cup \Phi')_{x=i})| \\ &= 2^{|R_{x=i}(\Phi)| + |R_{x=i}(\Phi')|} |M(\Phi_{x=i} \cup \Phi'_{x=i})| \\ &= 2^{|R_{x=i}(\Phi)|} 2^{|R_{x=i}(\Phi')|} |M(\Phi_{x=i})| |M(\Phi'_{x=i})| \\ &= |M_{x=i}(\Phi)| |M_{x=i}(\Phi')|. \end{aligned}$$

Por lo tanto

$$\begin{aligned} \#\text{sat}(\Phi \cup \Phi', x) &= \\ &= (|M_{x=1}(\Phi)| |M_{x=1}(\Phi')|, |M_{x=0}(\Phi)| |M_{x=0}(\Phi')|) \\ &= (|M_{x=1}(\Phi)|, |M_{x=0}(\Phi)|) \odot (|M_{x=1}(\Phi')|, |M_{x=0}(\Phi')|) \\ &= \#\text{sat}(\Phi, x) \odot \#\text{sat}(\Phi', x) \end{aligned}$$

Se tiene un lema análogo para el caso de las cargas conjuntas de dos variables respecto a la unión de dos fórmulas que tienen sólo una variable en común (ver figura 2).

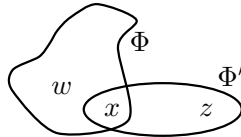


Fig. 2. Ubicación de las variables w y z con respecto a la unión de Φ y Φ'

LEMA 2.5. Si $\text{Var}(\Phi) \cap \text{Var}(\Phi') = \{x\}$, $w \in \text{Var}(\Phi) \setminus \text{Var}(\Phi')$ y $z \in \text{Var}(\Phi') \setminus \text{Var}(\Phi)$ entonces

$$\#\text{sat}(\Phi \cup \Phi', w, z) = \#\text{sat}(\Phi, w, x) \#\text{sat}(\Phi', x, z).$$

DEMOSTRACIÓN. Para simplificar la notación, es cómodo suponer que $a_{ij} = |M_{w=i, x=j}(\Phi)|$, $b_{ij} = |M_{x=i, z=j}(\Phi')|$ y $c_{ij} = |M_{w=i, z=j}(\Phi \cup \Phi')|$, esto es las entradas de $\#\text{sat}(\Phi, w, x)$, $\#\text{sat}(\Phi', x, z)$ y $\#\text{sat}(\Phi \cup \Phi', w, z)$ son a_{ij} , b_{ij} y c_{ij} respectivamente. De la observación 1 se tiene que

$$\begin{aligned} M_{w=i, z=j}(\Phi \cup \Phi') &= \\ M_{w=i, x=1, z=j}(\Phi \cup \Phi') \cup M_{w=i, x=0, z=j}(\Phi \cup \Phi') \end{aligned}$$

como $\Phi_{x=k}$ y $\Phi'_{x=k}$ no tienen variables en común para cada $k \in \{0, 1\}$, de (6) se tiene que

$$|M_{w=i, x=k, z=j}(\Phi \cup \Phi')| = a_{ik} b_{kj},$$

de lo anterior es claro que

$$c_{ij} = a_{i1} b_{1j} + a_{i0} b_{0j}$$

3. Operador para conteo de cadenas

Se comienza con la definición formal de los conceptos de hipergrafo, cadena y ciclo.

DEFINICIÓN 3.1. Un hipergrafo G es un par de conjuntos disjuntos (V, E) , donde V es conjunto no vacío, llamado conjunto de vértices o nodos y los elementos de E (hiperaristas de G), son subconjuntos de V . Un grafo es un hipergrafo donde todas sus hiperaristas tienen dos elementos.

DEFINICIÓN 3.2. Una *cadena* es un hipergrafo $P = (V, E)$ con $E = \{e_1, \dots, e_m\}$, donde las hiperaristas e_i son todas distintas, $e_i \cap e_{i+1} \neq \emptyset$ y $e_i \cap e_j = \emptyset$ si $1 < |i - j|$.

3.1. Cadenas simples

También, como en el caso de grafos, se acostumbra denotar a $P = e_1 e_2 \dots e_m$ como un camino en el hipergrafo. Un ciclo es un hipergrafo de la forma $C = e_1 e_2 \dots e_m + e$ donde e es una hiperarista que no pertenece a las hiperaristas de P y $e \cap e_1 \neq \emptyset \neq e \cap e_m$.

El hipergrafo de restricciones de una fórmula Φ en CNF representa una cadena simple si sus cláusulas pueden ser ordenadas de tal manera que dos cláusulas consecutivas coincidan en una variable. Es decir, podemos escribir $\Phi = \{c_1, c_2, \dots, c_m\}$, donde $|c_i \cap c_{i+1}| = 1$, para $i \in \{1, \dots, m-1\}$ y $c_i \cap c_j = \emptyset$ con $1 < |i - j|$. La fórmula Φ es llamada también cadena simple y se denota por $\Phi = c_1 c_2 \dots c_m$ (ver figura 3).

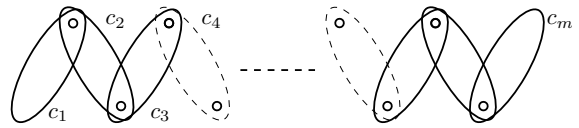


Fig. 3. Hipergrafo: cadena simple Φ

Mediante una tabla de verdad podemos identificar un operador de conteo para fórmulas cuyos hipergrafos representan cadenas simples. Este operador se propone en la siguiente definición.

DEFINICIÓN 3.3. Sea $\mathbb{S} : \mathbb{N}^2 \rightarrow \mathbb{N}^2$ el operador definido por

$$\mathbb{S} = \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix}$$

\mathbb{S} es llamado *operador arista*.

El siguiente lema establece como se calcula la carga cuando agregamos una hiperarista a una fórmula arbitraria que toca sólo un nodo de ésta (ver figura 4).

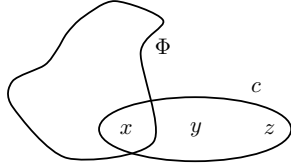


Fig. 4. Fórmula Φ con la hiperarista c que coincide en la variable x

LEMA 3.4. Sea Φ una fórmula y $c = \{x, y, z\}$ una cláusula tal que $Var(\Phi) \cap c = \{x\}$, entonces

$$\#\text{sat}(\Phi \cup c, z) = \#\text{sat}(\Phi \cup c, y) = \mathbb{S}\#\text{sat}(\Phi, x),$$

donde \mathbb{S} es el operador arista.

DEMOSTRACIÓN. Es claro que Φ y c satisfacen las condiciones del lema 2.4. Entonces

$$\#\text{sat}(\Phi \cup c, x) = \#\text{sat}(\Phi, x) \odot \#\text{sat}(c, x)$$

Por otra parte tenemos que si $\#\text{sat}(\Phi, x) = (\alpha, \beta)$ y como $\#\text{sat}(c, x) = (4, 3)$, entonces

$$\#\text{sat}(\Phi \cup c, x) = (4\alpha, 3\beta).$$

En consecuencia, $|M_{x=1}(\Phi)| = 4\alpha$ y $|M_{x=0}(\Phi)| = 3\beta$.

Cada modelo $s \in M_{x=1}(\Phi \cup c)$, toma dos posibles valores en z (en y), $s(z) = 0$ ó $s(z) = 1$. Entonces

$$|M_{x=1, z=1}(\Phi \cup c)| = 2\alpha \quad y \quad |M_{x=1, z=0}(\Phi \cup c)| = 2\alpha.$$

Por otro lado, para cada modelo $s \in M_{x=0}(\Phi \cup c)$ se tienen tres posibilidades para los valores de s en las variables y y z , estos son: $s(y) = 1 \wedge s(z) = 1$, $s(y) = 1 \wedge s(z) = 0$ y $s(y) = 0 \wedge s(z) = 1$. Esto es, del total de modelos en $M_{x=0}(\Phi \cup c)$, en $2/3$ partes se tiene que s toma el valor de uno en la variable z y en la parte restante s toma el valor de cero en esta variable. En consecuencia

$$|M_{x=0, z=1}(\Phi \cup c)| = 2\beta \quad y \quad |M_{x=0, z=0}(\Phi)| = \beta.$$

Se sigue que la carga de z relativa a la fórmula Φ es

$$\begin{aligned} \#\text{sat}(\Phi \cup c, z) &= (|M_{z=1}(\Phi \cup c)|, |M_{z=0}(\Phi \cup c)|) \\ &= (2\alpha + 2\beta, 2\alpha + \beta) \\ &= \mathbb{S}\#\text{sat}(\Phi, x) \end{aligned}$$

de manera análoga se obtiene $\#\text{sat}(\Phi \cup c, y) = \mathbb{S}\#\text{sat}(\Phi, x)$.

El siguiente teorema establece cómo contar el número de modelos de una fórmula cuyo hipergrafo asociado es una cadena simple.

TEOREMA 3.5. Sea $\Phi_m = c_1 c_2 \cdots c_m$, una cadena simple en 2μ -3MON. Suponga que $c_m = \{x, y, z\}$ y $x \in c_m \cap c_{m-1}$, entonces

$$\#\text{sat}(\Phi_m, z) = \#\text{sat}(\Phi_m, y) = \mathbb{S}^m \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad (7)$$

y si $w \in c_1 \setminus c_2$, entonces

$$\#\text{sat}(\Phi_m, w, z) = \mathbb{S}^m, \quad (8)$$

donde \mathbb{S} es el operador arista.

DEMOSTRACIÓN. Por inducción sobre m . El caso base $m = 1$ es trivial. Suponer que el teorema es válido para $m - 1$. Considere la cadena $\Phi_{m-1} = c_1 c_2 \cdots c_{m-1}$. Es claro que Φ_{m-1} y c_m satisfacen las condiciones del lema 3.4. Entonces

$$\#\text{sat}(\Phi_m, z) = \#\text{sat}(\Phi_{m-1} \cup c_m, z) = \mathbb{S}\#\text{sat}(\Phi_{m-1}, x)$$

Por hipótesis inductiva se tiene

$$\#\text{sat}(\Phi_{m-1}, x) = \mathbb{S}^m \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

de donde se sigue (7). Por otra parte, del lema 2.5 se tiene

$$\begin{aligned} \#\text{sat}(\Phi_m, w, z) &= \#\text{sat}(\Phi_{m-1} \cup c_m, w, z) \\ &= \#\text{sat}(\Phi_{m-1}, w, x) \#\text{sat}(c_m, x, z) \\ &= \mathbb{S}^{m-1} \mathbb{S}, \end{aligned}$$

de donde se sigue el teorema.

EJEMPLO 3.6. Ilustremos el paso inductivo del teorema previo. Supongamos que $c = \{u, v, x\}$ y $c' = \{x, y, z\}$ son dos cláusulas. Si (m, n) es la carga de la variable x entonces por cada uno de los m modelos las variables y y z pueden tomar los valores de 0 o 1 en la extensión respectiva, para que c' sea verdadera, similarmente, para cada uno de los n modelos, la variable y puede tomar el valores de 0 o 1, sin embargo, en las extensiones donde la variable y tome el valor de 0, la z tendrá que tomar el valor de 1 (ver figura 5).

Entonces hay $2m + 2n$ extensiones con el valor de 1 en la variable z y $2m + n$ con el valor de 0 en esta misma variable, esto es la carga de z relativa a $c \cup c'$ es $\mathbb{S} \begin{pmatrix} m \\ n \end{pmatrix} = \mathbb{S}\#\text{sat}(c, x)$.

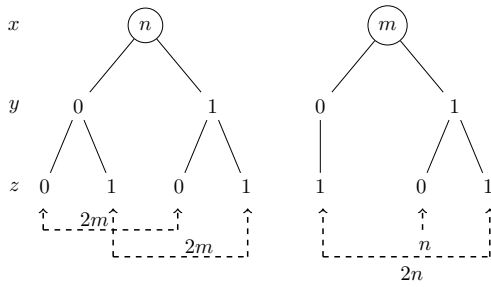


Fig. 5. Árbol de posibilidades del ejemplo 3.6

Dados dos hipergrafos con un sólo nodo en común, por el lema 2.4 podemos calcular la carga del nodo con respecto a la unión de los hipergrafos. Esto significa que si tenemos la carga del nodo relativa a cada una de las subestructuras, entonces podemos conocer la carga del nodo en la estructura completa. Por lo que, un punto importante cuando se analiza el conteo de modelos basado en subestructuras es calcular la distribución de cargas de los nodos en cada subestructura, o por lo menos de aquellos nodos que nos ayudan a relacionar una subestructura con otra. Por ejemplo, si se analiza cómo se da la distribución de las cargas de las variables y, z de la cláusula $c = \{x, y, z\}$ conociendo la carga de la variable x con respecto a una fórmula Φ tal que $Var(\Phi) \cap c = \{x\}$ tenemos del lema 3.4 que

$$\#\text{sat}(\Phi \cup c, y) = \#\text{sat}(\Phi \cup c, z) = \mathbb{S}\#\text{sat}(\Phi, x),$$

Para ejemplificar la discusión anterior, se calcula la distribución de cargas de y, z con respecto a la fórmula $\Phi = \{c_1, c_2, c_3\}$, donde $c_1 = \{\alpha, \beta, x\}$, $c_2 = \{\gamma, \delta, x\}$ y $c_3 = \{x, y, z\}$ (ver figura 6).

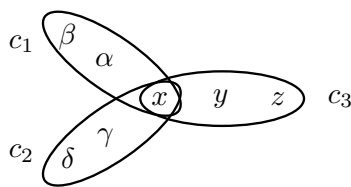


Fig. 6. Hipergrafo de la fórmula $\Phi = \{c_1, c_2, c_3\}$

Del lema 2.4 tenemos

$$\begin{aligned} \#\text{sat}(c_1 \cup c_2, x) &= \#\text{sat}(c_1, x) \odot \#\text{sat}(c_2, x) \\ &= \binom{4}{3} \odot \binom{4}{3} = \binom{16}{9} \end{aligned}$$

y del lema 3.4

$$\begin{aligned} \#\text{sat}(c_1 c_2 \cup c_3, y) &= \#\text{sat}(c_1 c_2 \cup c_3, z) \\ &= \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 16 \\ 9 \end{pmatrix} = \begin{pmatrix} 50 \\ 41 \end{pmatrix} \end{aligned}$$

por lo tanto, hay 91 modelos que satisfacen la fórmula Φ . Observe que en este caso la fórmula Φ no es una cadena.

3.2. Cadenas alternadas

En esta sección se analiza el conteo de modelos para fórmulas en 2 μ -3MON cuyos hipergrafos corresponden a cadenas “alternadas”. La siguiente definición aclara este concepto.

DEFINICIÓN 3.7.

- Dos cláusulas c, c' en una fórmula, se dice que están *simplemente ligadas* si y sólo si $|c \cap c'| = 1$.
- Las cláusulas c, c' están *doblemente ligadas* si y sólo si $|c \cap c'| = 2$.
- Una fórmula Φ es una *cadena alternada* si podemos escribir $\Phi = c_1 c_2 \cdots c_m$ donde
 - c_i y c_{i+1} están simple o doblemente ligadas,
 - c_{i-1} y c_{i+1} no pueden estar doblemente ligadas a c_i al mismo tiempo,
 - $c_i \cap c_j = \emptyset$ si $|i - j| > 1$.

EJEMPLO 3.8. La fórmula $\Phi = c_1 c_2 \cdots c_5$, donde $c_1 = \{a, b, c\}$, $c_2 = \{b, c, d\}$, $c_3 = \{d, e, f\}$, $c_4 = \{f, g, h\}$ y $c_5 = \{g, h, i\}$ es una cadena alternada (ver figura 7).

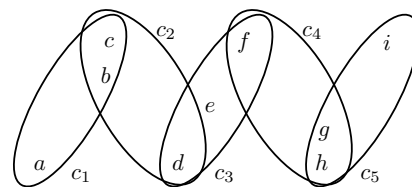


Fig. 7. Hipergrafo representando una cadena alternada del ejemplo 3.8

Del análisis de la estructura más simple de una cadena alternada se determina un operador de conteo de modelos que actúa sobre estructuras que corresponden a cláusulas doblemente ligadas. En la siguiente definición se detalla este operador.

DEFINICIÓN 3.9. El operador $\mathbb{D} : \mathbb{N}^2 \rightarrow \mathbb{N}^2$ definido por

$$\mathbb{D} = \begin{pmatrix} 1 & 1 \\ 3/2 & 0 \end{pmatrix}$$

es llamado operador *doble-arista*.

El operador \mathbb{D} aplicado a la carga de cualquier variable de una cadena alternada siempre produce cargas con entradas enteras. En efecto, como una cadena alternada en 2μ -3MON no puede tener dos cláusulas consecutivas doblemente ligadas, cuando aplicamos el operador \mathbb{D} a una carga (m, n) , hay dos posibilidades, que (m, n) sea una carga inicial, esto es $(m, n) = (4, 3)$, ó que la carga (m, n) provenga de la aplicación del operador arista \mathbb{S} a otra carga (m', n') , es decir $(m, n) = \mathbb{S}(m', n') = (2m' + 2n', 2m' + n')$. En cualquier caso $\mathbb{D}(m, n)$ tiene entradas enteras.

En el siguiente lema se establece cómo calcular la carga de una variable con respecto a la unión de una fórmula Φ con una cláusula c que coincide en dos variables con dicha fórmula (ver figura 8).

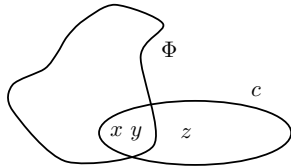


Fig. 8. Fórmula que coincide en dos variables con una cláusula

LEMA 3.10. Sea Φ una fórmula y $c = \{x, y, z\}$ una cláusula tal que $\text{Var}(\Phi) \cap c = \{x, y\}$, entonces

$$1. \# \text{sat}(\Phi \cup c, z) = (|M(\Phi)|, |M(\Phi)| - a_{00})$$

$$2. \# \text{sat}(\Phi \cup c, x) = (\mathbb{S} \odot \# \text{sat}(\Phi, x, y)) \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

DEMOSTRACIÓN. De la definición de carga conjunta se tiene que $m_{ij} = |M_{x=i, y=j}(\Phi)|$ son las entradas de la matriz $\# \text{sat}(\Phi, x, y)$. Si z tiene el valor de 1 bajo una extensión de un modelo en $M(\Phi)$, la cláusula c es válida sin importar los valores que tomen las variables x e y , de aquí que existen $m_{11} + m_{10} + m_{01} + m_{00} = |M(\Phi)|$ modelos en $M(\Phi \cup c)$, donde z toma el valor de 1. Si z tiene el valor de 0, bajo otra extensión de un modelo en $M(\Phi)$, entonces bajo esta misma extensión alguna variable de $\{x, y\}$ debe tener el valor de 1 para que la cláusula c sea válida, en consecuencia hay

$m_{11} + m_{10} + m_{01} = |M(\Phi)| - m_{00}$ modelos de $M(\Phi \cup c)$ donde z tiene el valor de 0, de aquí obtenemos la parte 1 del lema. Para la parte 2, de lo anterior se sigue que en $M(\Phi \cup c)$ hay $2m_{11} + 2m_{10}$ donde la variable x toma el valor de 1 y $2m_{01} + m_{00}$ donde toma el valor de 0, realizando el producto de Hadamard $\mathbb{S} \odot \# \text{sat}(\Phi, x, y)$ se llega a la igualdad de la parte 2 del lema.

El siguiente teorema determina el conteo de modelos para cadenas alternadas.

TEOREMA 3.11. Sea $\Phi_m = c_1 c_2 \dots c_m$ una cadena alternada en 2μ -3MON y $\alpha \in c_m \setminus c_{m-1}$, entonces $\# \text{sat}(\Phi_m, \alpha)$ se obtiene aplicando la ecuación de recurrencia

$$\begin{cases} \mathbf{q}_1 = (4, 3); \\ \mathbf{q}_i = \Delta \mathbf{q}_{i-1}, \quad i = 2, \dots, m \end{cases} \quad (9)$$

donde $\mathbf{q}_1 = \# \text{sat}(\Phi_1, \alpha_1)$, $\alpha_1 \in c_1$, $\mathbf{q}_i = \# \text{sat}(\Phi_i, \alpha)$ con $\alpha_i \in c_i \setminus c_{i-1}$ y el operador Δ es definido por

$$\Delta = \begin{cases} \mathbb{S} & \text{si } c_i \text{ y } c_{i-1} \text{ están simplemente} \\ & \text{ligadas} \\ \mathbb{D} & \text{de otra manera} \end{cases}$$

El siguiente ejemplo nos sirve para ilustrar este teorema.

EJEMPLO 3.12. Considere la cadena alternada dada en el ejemplo 3.8. Utilizando la ecuación de recurrencia (9) se tiene

$$\begin{aligned} \mathbf{q}_1 &= \begin{pmatrix} 4 \\ 3 \end{pmatrix} \\ \mathbf{q}_2 &= \begin{pmatrix} 1 & 1 \\ 3/2 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6 \end{pmatrix} \\ \mathbf{q}_3 &= \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 7 \\ 6 \end{pmatrix} = \begin{pmatrix} 26 \\ 20 \end{pmatrix} \\ \mathbf{q}_4 &= \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 26 \\ 20 \end{pmatrix} = \begin{pmatrix} 92 \\ 72 \end{pmatrix} \\ \mathbf{q}_5 &= \begin{pmatrix} 1 & 1 \\ 3/2 & 0 \end{pmatrix} \begin{pmatrix} 92 \\ 72 \end{pmatrix} = \begin{pmatrix} 164 \\ 138 \end{pmatrix} \end{aligned}$$

donde \mathbf{q}_1 es la carga de las variables a, b y c .
 \mathbf{q}_2 es la carga de la variable d .
 \mathbf{q}_3 es la carga de e y f .

q_4 es la carga de las variables g y h .

q_5 es la carga de la variable i .

Por lo tanto, existen 302 modelos que satisfacen a la fórmula Φ .

DEMOSTRACIÓN (Teorema 3.11). El caso base $m = 1$, es trivial. Para cada $i = 2, 3, \dots, m$ existen dos casos:

Caso 1. c_{i-1} y c_i simplemente ligadas. Suponga que $x \in c_i \setminus c_{i-1}$ y que $c_i = \{x, y, z\}$, del lema 3.4 se sigue directamente que $q_i = \mathbb{S}q_{i-1}$.

Caso 2. c_{i-1} y c_i doblemente ligadas. Entonces se tiene que c_{i-1} y c_{i-2} deben estar simplemente ligadas. Supongamos $c_{i-1} = \{w, x, y\}$ y $c_i = \{x, y, z\}$ (ver figura 9).

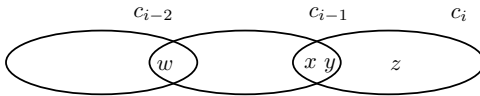


Fig. 9. Hipergrafo: cadena alternada, caso 2 del teorema 3.11

Suponga también que (ℓ_1, ℓ_2) es la carga de $w \in c_{i-1} \setminus c_{i-2}$ relativa a la cadena Φ_{i-2} . Por el caso 1, la carga de w relativa a Φ_{i-1} es $(2\ell_1 + 2\ell_2, 2\ell_1 + \ell_2)$, esto es $|M(\Phi_{i-1})| = 4\ell_1 + 3\ell_2$. Por el lema 3.10 parte 1, se tiene que

$$q_i = \#\text{sat}(\Phi_{i-1} \cup c_i, z)$$

$$= (|M(\Phi_{i-1})|, |M(\Phi_{i-1})| - a_{00}),$$

donde a_{00} es el número de modelos de Φ_{i-1} y x e y toman el valor de 0. Por otra parte se tiene que $|M(\Phi_{i-1})| = 4\ell_1 + 3\ell_2$ y

$$\begin{aligned} |M(\Phi_{i-1})| - a_{00} &= |M_{z=0}(\Phi_{i-1} \cup c_i)| \\ &= |M_{x \neq 0 \vee y \neq 0}(\Phi_{i-1})| \\ &= 3|M(\Phi_{i-2})| = 3\ell_1 + 3\ell_2 \end{aligned}$$

Por lo tanto,

$$q_i = (4\ell_1 + 3\ell_2, 3\ell_1 + 3\ell_2) = \mathbb{D}(2\ell_1 + 2\ell_2, 2\ell_1 + \ell_2)$$

$$= \mathbb{D}q_{i-1}$$

es la carga de z .

4. Conteo para ciclos

En esta sección se distinguen dos tipos de ciclos en 2μ -3MON, los ciclos simples y los ciclos simples alternados. Un hipergrafo de la forma $\Phi_m = c_1c_2 \cdots c_{m-1} + c_m$ es un *ciclo simple* si $c_1c_2 \cdots c_{m-1}$ es una cadena simple y $|c_1 \cap c_m| = 1$, y es un *ciclo alternado* si $c_1c_2 \cdots c_{m-1}$ es una cadena alternada y $1 \leq |c_1 \cap c_m| \leq 2$. A continuación realizamos el conteo de modelos para fórmulas cuyos hipergrafos representan ciclos simples.

4.1. Ciclos simples

Del análisis de estructuras simples y con la ayuda del producto de Hadamard se llega al siguiente teorema.

TEOREMA 4.1. Sea $\Phi_m = c_1c_2 \cdots c_{m-1} + c_m$ un ciclo simple en 2μ -3MON, entonces

$$\#\text{sat}(\Phi_m, \alpha) = (\mathbb{S} \odot \mathbb{S}^{m-1}) \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

donde $\alpha \in c_1 \cap c_m$ y \mathbb{S} es el operador arista.

Antes de presentar la prueba se ilustra el teorema con el ejemplo dado a continuación.

EJEMPLO 4.2. Sea

$\Phi = \{\{a, b, c\}, \{c, d, e\}, \{e, f, g\}, \{g, h, i\}, \{i, j, a\}\}$ se sigue del teorema previo que:

$$\begin{aligned} \#\text{sat}(\Phi, a) &= (\mathbb{S} \odot \mathbb{S}^4) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \left(\begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix} \odot \begin{pmatrix} 100 & 78 \\ 78 & 61 \end{pmatrix} \right) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 200 & 156 \\ 156 & 61 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 356 \\ 217 \end{pmatrix} \end{aligned}$$

Por lo tanto, existen 573 modelos para la fórmula Φ .

DEMOSTRACIÓN (Teorema 4.1). Supongase que $\beta \in c_{m-1}$, $c_m = \{\alpha, \beta, \gamma\}$ y que Φ_{m-1} denota la cadena simple $c_1c_2 \cdots c_{m-1}$. Del teorema 3.5 ecuación (8) y lema 3.10 parte 2, se siguen las igualdades:

$$\begin{aligned} \#\text{sat}(\Phi_m, \alpha) &= \#\text{sat}(\Phi_{m-1} \cup c_m, \alpha) \\ &= (\mathbb{S} \odot \#\text{sat}(\Phi_{m-1}, \alpha, \beta)) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= (\mathbb{S} \odot \mathbb{S}^{m-1}) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{aligned}$$

■

4.2. Ciclos alternados

Se tiene un teorema para ciclos alternados análogo al teorema 4.1 de ciclos simples. Antes de enunciar el teorema se establece la notación y terminología que sigue.

Dada la cadena alternada $c_1c_2 \cdots c_m$, se define $\Delta_1 = \mathbb{S}$, donde \mathbb{S} es el operador de cadena simple, y para $i \in \{2, 3, \dots, m\}$ se define

$$\Delta_i = \begin{cases} \mathbb{S} & \text{si } c_i \text{ y } c_{i-1} \text{ están simplemente} \\ & \text{ligadas} \\ \mathbb{D} & \text{de otra manera} \end{cases}$$

Ω_k para $k \in \{1, 2, \dots, m\}$, denota el producto $\prod_{i=1}^k \Delta_i$. Y π_1, π_2 denotan las matrices

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \text{ y } \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \text{ respectivamente.}$$

TEOREMA 4.3. Sea $\Phi_m = c_1c_2 \cdots c_{m-1} + c_m$ un ciclo alternado en 2μ -3MON y $\alpha \in c_1 \cap c_m$, entonces

$$\#\text{sat}(\Phi, \alpha) = \mathbb{S} \odot \Omega_{m-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

si c_m y c_{m-1} son simplemente ligadas y

$$\#\text{sat}(\Phi, \alpha) = (\Omega_{m-1} - \pi_1 \Omega_{m-2} \pi_2) \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

de otra manera.

Bajo el mismo esquema de las secciones previas, presentamos un ejemplo.

EJEMPLO 4.4. Sea $\Phi = \{\{a, b, c\}, \{c, d, e\}, \{d, e, a\}\}$, de acuerdo al teorema anterior, el número $\#\text{sat}(\Phi, a)$ se determina mediante:

$$\begin{aligned} \#\text{sat}(\Phi, a) &= (\Omega_2 - \pi_1 \Omega_0 \pi_1) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= (\mathbb{S}\mathbb{S} - \pi_1 \mathbb{S} \pi_2) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \left[\begin{pmatrix} 8 & 6 \\ 6 & 5 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix} \right] \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 8 & 6 \\ 6 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 14 \\ 9 \end{pmatrix} \end{aligned}$$

Entonces, la fórmula Φ tiene un total de 23 modelos (ver figura 10).

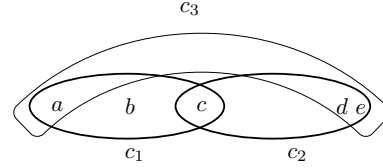


Fig. 10. Ciclo alternado, ejemplo 4.4

DEMOSTRACIÓN (Teorema 4.3). Sean Φ_{m-2} y Φ_{m-1} las cadenas $c_1 \cdots c_{m-2}$ y $c_1 \cdots c_{m-1}$ respectivamente. El ciclo alternado Φ_m se puede expresar en términos de las cadenas como

$$\Phi_m = \Phi_{m-1} + c_m = \Phi_{m-2} + c_{m-1} + c_m$$

Del teorema 3.5, se obtiene que

$$\begin{aligned} \#\text{sat}(\Phi_{m-1}, \alpha) &= \Omega_{m-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} m_{11} & m_{10} \\ m_{01} & m_{00} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{aligned}$$

y

$$\begin{aligned} \#\text{sat}(\Phi_{m-2}, \alpha) &= \Omega_{m-2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} m'_{11} & m'_{10} \\ m'_{01} & m'_{00} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{aligned}$$

donde, de la definición 2.2 y la observación 1, se debe tener que

$$m_{i,j} = |M_{\alpha=i, \gamma=j}(\Phi_{m-1})| \text{ y } m'_{i,j} = |M_{\alpha=i, \delta=j}(\Phi_{m-2})|.$$

Por otra parte se tienen dos casos:

Caso 1 (c_m y c_{m-1} simplemente ligadas). La prueba se sigue paso a paso de la demostración del teorema 4.1 sustituyendo \mathbb{S}^m por Ω_{m-1} .

Caso 2 (c_m y c_{m-1} doblemente ligadas). Se observa que c_{m-1} y c_{m-2} deben estar simplemente ligadas, ya que c_m y c_{m-1} son doblemente ligadas. Suponga que $\delta \in c_{m-2} \cap c_{m-1}$ y que $c_{m-1} = \{\delta, \beta, \gamma\}$, entonces $c_m = \{\alpha, \beta, \gamma\}$ (ver figura 11).

Puesto que la cláusula c_m es una restricción que debe cumplir un modelo de Φ_{m-1} que sea también modelo de Φ_m , se tiene la contención $M(\Phi_m) \subseteq M(\Phi_{m-1})$.

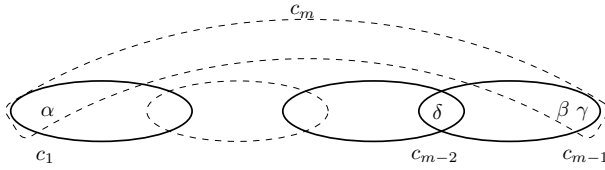


Fig. 11. Ciclo alternado Φ_m (demostración del teorema 4.3)

Por otro lado, se tienen las siguientes igualdades

$$\begin{aligned} |M_{\alpha=1,\gamma=1}(\Phi_m)| &= m_{11}, \\ |M_{\alpha=1,\gamma=0}(\Phi_m)| &= m_{10}, \\ |M_{\alpha=0,\gamma=1}(\Phi_m)| &= m_{01} \end{aligned}$$

ya que las asignaciones $\alpha = 1 \wedge \gamma = 1$, $\alpha = 1 \wedge \gamma = 0$ y $\alpha = 0 \wedge \gamma = 1$ hacen válida la cláusula c_m .

También es claro que

$$\begin{aligned} M_{\alpha=0,\gamma=0}(\Phi_m) &= M_{\alpha=0,\beta=1}(\Phi_{m-1}) \\ &= M_{\alpha=0,\delta=0}(\Phi_{m-2}) \\ &\cup M_{\alpha=0,\delta=1}(\Phi_{m-2}) \end{aligned}$$

por lo consiguiente se tiene

$$|M_{\alpha=0,\gamma=0}(\Phi_m)| = m'_{00} + m'_{01}$$

Ahora, como c_{m-1} y c_{m-2} están simplemente ligadas, del teorema 3.11 se cumple $\Omega_{m-1} = \mathbb{S}\Omega_{m-2}$, de donde se tiene la relación

$$m_{00} = m'_{00} + 2m'_{01}$$

Por lo tanto se tiene

$$\begin{aligned} \#sat(\Phi_m, \alpha) &= \begin{pmatrix} m_{11} & m_{10} \\ m_{01} & m'_{00} + m'_{01} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} m_{11} & m_{10} \\ m_{01} & m_{00} - m'_{01} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \left[\begin{pmatrix} m_{11} & m_{10} \\ m_{01} & m_{00} \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & m'_{01} \end{pmatrix} \right] \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{aligned}$$

El teorema se sigue de la igualdad

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} m'_{11} & m'_{10} \\ m'_{01} & m'_{00} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & m'_{01} \end{pmatrix}$$

5. Discusión

La implementación de todos los casos expuestos aquí, nos conducen a algoritmos tratables, ya que dada una fórmula Φ en CNF, identificar si pertenece a una de las clases estudiadas se puede hacer en tiempo cuadrático, esto se logra implementando un algoritmo de búsqueda a profundidad. Por otro lado, como se puede observar en todos los casos estudiados, el cálculo del número de modelos se reduce a la multiplicación de a lo más m matrices de 2×2 , donde m es el número de cláusulas de la fórmula en cuestión, por lo que la multiplicación de matrices se puede hacer en tiempo lineal con respecto a m , que es el tamaño de la fórmula de entrada.

El problema que nos hemos planteado es identificar subestructuras tratables de la clase $\#2\mu$ -3MON mediante el uso de operadores matriciales; las clases identificadas aquí corresponden a estructuras de ancho de hiper-árbol acotado por lo que, utilizando los resultados obtenidos por Fisher [8], existen algoritmos que realizan el conteo en estas clases en tiempo $O(n^3)$ sobre el número de cláusulas de la fórmula en contraste con los métodos mostrados aquí que realizan este conteo en tiempo $O(n)$.

Hasta donde sabemos la clase $\#2\mu$ -3MON no ha sido ubicada siguiendo la dirección de medidas acotadas de invariantes hipergráficos, como es el caso del ancho de hiper-árbol. Por esta razón hemos elegido aproximarnos a la ubicación de la clase estudiada a través de los operadores matriciales presentados en este trabajo.

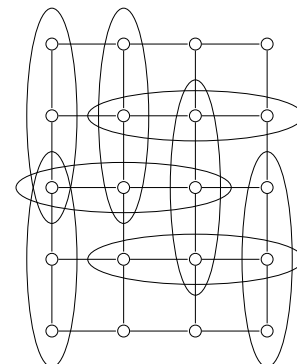


Fig. 12. Fórmula con estructura de pseudo-rejilla

6. Conclusiones

Se obtiene una subclase tratable de la clase sintáctica $\#2_{\mu}$ -3MON mediante métodos basados en operadores matriciales de conteo que actúan sobre estructuras de cadenas simples, cadenas alternadas, ciclos simples y ciclos alternados del hipergrafo asociado a la fórmula.

Como trabajo futuro, en la dirección de la identificación de estructuras tratables en hipergrafos, se pueden reconocer nuevas estructuras si se lleva la idea de los métodos de transferencia matricial aplicadas a grafos a hipergrafos [1]. En cuanto a la determinación de la ubicación de la clase $\#2_{\mu}$ -3MON dentro de las clases de complejidad, consideramos que es posible determinarla si logramos disponer el hipergrafo de una fórmula de la clase 2_{μ} -3MON en una estructura de pseudo-rejilla, esto es, las hiperaristas colocadas vertical u horizontalmente coinciden con las aristas de un grafo rejilla (ver figura 12).

Agradecimientos

Agradecemos a los revisores sus valiosas sugerencias y comentarios. Consideramos que fueron de gran utilidad para el enriquecimiento de nuestro trabajo.

Referencias

1. **Bautista, C. & Guillén, C. (2012).** Fibonacci numbers of generalized zykov sums. *Journal of Integer Sequences*, 1–22.
2. **Bubley, R. (2001).** Randomized algorithms: Approximation, generation, and counting. In *Distinguished dissertations*. Springer, 1–20.
3. **Bubley, R. & Dyer, M. (1997).** Graph orientations with no sink and an approximation for a hard case of $\#sat$. In *Proceedings of the third annual ACM symposium on Theory of computing*. 248–257.
4. **Cook, S. (1971).** The complexity of theorem proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*. 151–158.
5. **Courcelle, B., Makowsky, J., & Rotics, U. (2001).** On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics*, 23–52.
6. **Diestel, R. (2005).** *Graph Theory*. Springer-Verlag.
7. **Dyer, M. & Greenhil, L. (1997).** $\#p$ -completeness proofs for colourings and independent sets. In *Technical Report 97.47, School of Computer Studies, University of Leeds*.
8. **Fischer, E., Makowsky, J., & Ravve, E. (2008).** Counting truth assignments of formulas of bounded tree-width or clique-width. *Discrete Applied Mathematics*, 156(4), 511–529.
9. **Gottlob, G., Grohe, M., N., M., Samer, M., & F., S. (2005).** Hypertree decompositions: structure, algorithms, and applications. In *Proceedings of Graph-Theoretic Concepts in Computer Science*. LNCS 3787, 1–15.
10. **Mezey, P. (1989).** *Topological Methods in Chemistry*. John Wiley & Sons, Inc.
11. **Nebel, B. (1991).** Belief revision and default reasoning: Syntax-based approaches. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*. Cambridge, MA, USA, 417–428.
12. **Nishimura, N., Ragde, P., & Szeider, S. (2006).** Solving $\#sat$ using vertex covers. In *Proc. Ninth International Conference on Theory and Applications of Satisfiability Testing*. LNCS 4121, 396–409.
13. **Roth, D. (1996).** On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2), 273–302.
14. **Sinclair, A. (1993).** *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Cambridge, Massachusetts, Boston: Birkhäuser Verlag.
15. **Vadhan, S. (1995).** *The Complexity of Counting*. Harvard University, Cambridge, Massachusetts.
16. **Vadhan, S. (2001).** The complexity of counting in sparse, regular, and planar graphs. *SIAM J. Comput.*, 31(2), 398–427.
17. **Valiant, L. (1979).** The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3), 410–421.



Carlos Guillén obtuvo el grado de Doctor en Ciencias Computacionales por el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) y el grado de Maestría en Matemáticas por la Facultad de Físico Matemáticas de la BUAP. Es

profesor de tiempo completo en la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla desde 1995. Sus intereses de investigación son en teoría de la computación y procesamiento de imágenes.



Rafael Lemuz obtuvo el título de Licenciado en Ciencias de la Computación por la Benemérita Universidad Autónoma de Puebla en 2002. Obtuvo el grado de Maestro y Doctor en Ciencias de la Computación por el

Instituto Nacional de Astrofísica, Óptica y Electrónica en 2003 y 2008 respectivamente.

Desde el 2008 es profesor-investigador en la Benemérita Universidad Autónoma de Puebla. Su investigación se enfoca en el desarrollo de modelos matemáticos y computacionales para el procesamiento de señales e imágenes. La motivación del trabajo del Dr. Rafael Lemuz radica en aplicaciones de la visión computacional en biomedicina.



Irene Ayaquica obtuvo el grado de Doctor en Ciencias Computacionales por el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE); el grado de Maestro en Ciencias de la Computación por el Centro de Investigación en Computación

del IPN (CIC-IPN) y el título de Licenciado en Ciencias de la Computación por la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla (FCC-BUAP). Es profesor de tiempo completo en la Facultad de Computación de la Benemérita Universidad Autónoma de Puebla desde 2009. Sus principales líneas de investigación son reconocimiento de patrones y agrupamiento conceptual difuso.

Article received on 08/06/2012; accepted on 17/06/2013.