

## Algoritmos de conteo de las familias de Moore aplicación al caso $n = 7$

Pierre Colomb<sup>1</sup>, Olivier Raynaud<sup>2</sup>

*LIMOS, Universidad de Clermont–Ferrand, Francia*

Alexis Irlande<sup>3</sup>

*Departamento de Matemáticas*

*Universidad Nacional de Colombia*

*Bogotá*

*Una familia de Moore sobre un conjunto  $U_n = \{0, 1, \dots, n - 1\}$  es una colección de conjuntos  $\mathcal{M}$  cerrada para la operación de intersección y que contiene  $U_n$ . El conjunto de las familias de Moore para un  $n$  dado, notado  $\mathbf{M}_n$ , crece de forma más que exponencial con respecto a  $n$ , así  $|\mathbf{M}_3|$  vale 61 y  $|\mathbf{M}_4|$  vale 2480. En [9], los autores han determinado este número para  $n = 6$  en 24h. La evaluación de este número para  $n = 7$  es entonces un reto técnico difícil. En este artículo, presentamos una estrategia de conteo de las familias de Moore para  $n = 7$  y damos su valor: 14 087 648 235 707 352 472. Nuestro cálculo se apoya en particular sobre la enumeración de las familias de Moore equivalentes mediante un isomorfismo para  $n$  de 1 a 6.*

Palabras Claves: familias de Moore, conteo, simetría.

*A Moore family on a set  $U_n = \{0, 1, \dots, n - 1\}$  is as a collection of sets  $\mathcal{M}$  which is closed by intersection and containing  $U_n$ . The set of Moore families for a given  $n$ , denoted by  $\mathbf{M}_n$ , increases faster than exponentially with respect to  $n$ , thus  $|\mathbf{M}_3|$  is 61 and  $|\mathbf{M}_4|$  is 2480. In [9] the authors found this number for  $n = 6$  in 24h. Thus, its evaluation for  $n = 7$  can be considered as a difficult technical challenge. In this paper, we will introduce a counting strategy for Moore families in the case  $n=7$  and give its value: 14 087 648 235 707 352 472. Our calculation is mostly based on the enumeration of Moore families up to an isomorphism for  $n$  ranging from 1 to 6.*

Keywords: Moore Family, Enumeration, Symmetry.

MSC: 05–04, 05A15

---

<sup>1</sup> colomb@isima.fr

<sup>2</sup> raynaud@isima.fr

<sup>3</sup> airlande@unal.edu.co

## 1. Introduction

El conteo (y/o la enumeración) de un gran conjunto de objetos de naturaleza matemática es un desafío agradable de aceptar. Este tipo de ejercicio necesita proponer procesos algorítmicos originales combinando un conocimiento a profundidad de las propiedades de los objetos que hay que contar, de las estructuras de datos eficientes, pero también, y a menudo imperativamente, de las técnicas de programación avanzadas. En este artículo, nuestro esfuerzo se ha dirigido al conteo de las familias de Moore resultantes de un conjunto  $U_n = \{0, 1, \dots, n-1\}$  dado. La noción de familia de Moore, o de operador de clausura (función de  $2^{U_n}$  en  $2^{U_n}$ , extensiva, monótona e idempotente), o también de sistema de implicaciones, es común a un gran número de dominios. Citemos como ejemplo para las matemáticas los trabajos de [4] en álgebra, en informática citemos [5] en teoría de órdenes y de retículos y [6] en bases de datos relacionales. Mencionemos finalmente [7] y [8] para el análisis de datos. El mismo término de familia de Moore fue usado por primera vez por Birkhoff en [1] haciendo referencia a los trabajos del principio del siglo veinte de E.H. Moore en [12]. Técnicamente, una familia de Moore sobre  $U_n$  es una colección de conjuntos (o familia)  $\mathcal{M}$  cerrada para la operación de intersección y que contiene  $U_n$  (véase figura 1).

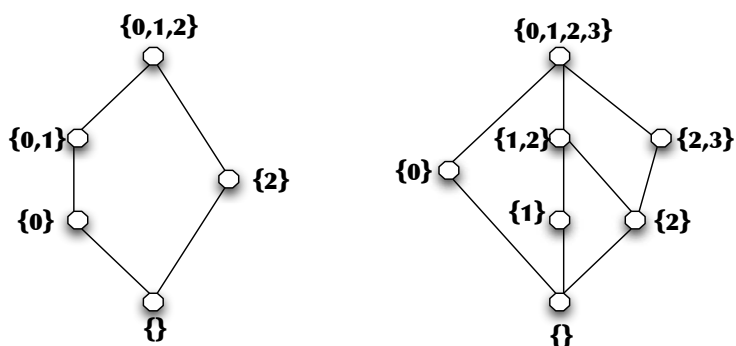


Figura 1: Dos familias de Moore: a la izquierda, la familia  $\{\emptyset, \{0\}, \{0, 1\}, \{2\}, \{0, 1, 2\}\}$  sobre el conjunto  $\{0, 1, 2\}$ , a la derecha, la familia  $\{\emptyset, \{0\}, \{1\}, \{2\}, \{1, 2\}, \{2, 3\}, \{0, 1, 2, 3\}\}$  sobre el conjunto  $\{0, 1, 2, 3\}$ .

El conjunto de las familias de Moore sobre  $U_n$ , notado  $M_n$ , es un sistema de clausura, es decir un conjunto de puntos fijos para cierto ope-

rador de clausura. Así, este sistema compuesto de las familias de Moore contiene un elemento máximo ( $2^{U_n}$ : el conjunto de las partes de  $U_n$ ) y la intersección de dos familias de Moore es ella misma una familia de Moore. Para una mirada global sobre las propiedades de este sistema de clausura, véase [3]. Al ordenar por inclusión el conjunto de los elementos de un sistema de clausura, obtenemos una estructura de retículo. Damos en la figura 2 el retículo de las familias de Moore para  $n = 2$ .

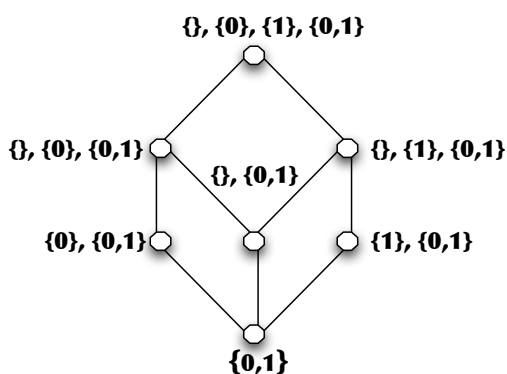


Figura 2: Diagrama de Hasse o retículo de las familias de Moore sobre  $U_2 = \{0, 1\}$ .

En [2], Burosch califica la cuestión del conteo de las familias de Moore sobre  $n$  de natural y propone una cuota superior a este número. Completaremos en el presente artículo su estudio, poniendo en evidencia una cota natural de  $|M_n|$  definida a partir de  $|M_{n-1}|$ . En 1998, Higuchi calculó en [10],  $|M_5|$  por un recorrido a profundidad del grafo de cobertura del retículo de las familias de Moore. Más recientemente, Habib y Nourine han calculado en [9] el cardinal de  $M_6$ . Su método se apoya sobre la existencia de una aplicación biyectiva entre el conjunto de las familias de Moore y el conjunto de los ideales coloreados de un orden compuesto de retículos booleanos. Gracias a un algoritmo eficiente de enumeración de los ideales de un orden (véase [11]), los autores logran censar  $M_6$  y anunciaron que el proceso necesita 24h sobre un Pentium III (600 MHz). El conjunto de los valores de estos conteos es dado en la tabla 1. La evolución exponencial de los resultados anunciaba la evaluación de este número para  $n = 7$  como un desafío técnico particularmente difícil.

La continuación del artículo se compone de lo siguiente. La sección 2 trata de las descripciones de la estructura de datos y de los puntos

$n$	$ M_n $	Referencia
0	1	
1	2	
2	7	
3	61	
4	2 480	
5	1 385 552	[10]
6	75 973 751 474	[9]
7	14 087 648 235 707 352 472	Este artículo

Cuadro 1: Valores conocidos de  $|M_n|$  para  $n \leq 7$ 

clave sobre los cuales se fundamenta nuestra estrategia de cálculo: **simetrías**, **forma canónica** y **familia máxima**. En la sección 3, damos los grandes principios algorítmicos usados. Luego, la sección 4 trata de los aspectos tecnológicos del cálculo (tipo de máquina, desempeño, índices de confiabilidad). Al final, concluimos con unas perspectivas.

## 2. Elementos de estrategia

Esta parte pretende establecer los elementos esenciales de nuestra estrategia de cálculo. Empezaremos detallando la codificación usada para almacenar y manipular las familias de Moore. Luego explicaremos diversos conceptos sobre los cuales se basa la estrategia de conteo: las simetrías entre familias, la noción de forma canónica como identificador de una clase de equivalencia y en fin la noción de familia máxima basada en la estructura recursiva de los objetos por contar. Las pruebas de las proposiciones son dadas en el anexo.

En la introducción, hemos definido una familia de Moore sobre  $U_n$  como una colección de conjuntos que contiene  $U_n$  y es cerrada por intersecciones. Sin embargo, por razones de comodidad que aparecerán al lector a lo largo del texto, todo nuestro procedimiento algorítmico será dedicado a la enumeración de las familias cerradas por unión y que contienen al conjunto vacío. Por supuesto, el conjunto de las familias de Moore corresponde a este conjunto por una aplicación biyectiva. En efecto, para una familia cerrada por unión que contiene el vacío, basta tomar el complemento de cada conjunto para obtener una familia de Moore (y vice-versa). Como ejemplo, a la familia de Moore sobre  $U_3$   $\{\{0\}, \{0, 1\}, \{0, 2\}, \{0, 1, 2\}\}$  corresponde la familia  $\{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$  y

recíprocamente.

### 2.1. Codificación

Consideremos un universo  $U_n = \{0, \dots, n - 1\}$  con  $n$  elementos y un conjunto  $E \subseteq U_n$ ,  $E$  puede ser codificado naturalmente por su vector característico (véase. figura 3 para un ejemplo).

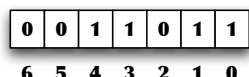


Figura 3: Sobre un universo con 7 elementos, el conjunto  $\{0, 1, 3, 4\}$  es codificado por un vector de 7 bits  $b[]$  donde  $b[i] = 1$  si y solamente si  $i \in \{0, 1, 3, 4\}$ .

Interpretando el vector característico como un número escrito en binario, se puede asociar a cada conjunto  $E$  un valor decimal. Más precisamente, a cada subconjunto de  $\{0, \dots, n - 1\}$  corresponde un entero entre 0 y  $2^n - 1$ . En el ejemplo anterior, el entero asociado al conjunto  $\{0, 1, 3, 4\}$  corresponde a la suma  $2^0 + 2^1 + 2^3 + 2^4$  es decir a 27. Usando estos valores decimales como identificadores de conjuntos, podemos codificar una familia de conjuntos sobre un universo con  $n$  elementos por un vector característico de  $2^n$  bits (véase figura 4 para un ejemplo).

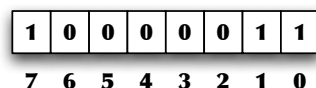


Figura 4: Sobre un universo con 3 elementos, la familia  $\mathcal{F} = \{\emptyset, \{0\}, \{0, 1, 2\}\}$  es codificada por un vector  $b[]$  de  $2^3 = 8$  bits donde  $b[i] = 1$  si y solamente si el conjunto  $E$  pertenece a  $\mathcal{F}$  ( $i$  siendo el entero asociado al conjunto  $E$ ).

Finalmente, interpretando una última vez el nuevo vector obtenido como un número escrito en binario, asociamos a cada familia un entero entre 0 y  $2^{2^n}$ . Como ejemplo, el identificador de la familia  $\{\emptyset, \{0\}, \{0, 1, 2\}\}$  es el entero 131. Como vamos a verlo, el uso de este código permite hacer las operaciones usuales sobre los conjuntos con operaciones lógicas o aritméticas sencillas.

En nuestro proceso de enumeración, dos operaciones son esenciales: comprobar si una familia contiene el conjunto vacío y comprobar si al agregar un conjunto  $E$  a una familia  $\mathcal{F}$  cerrada por unión,  $\mathcal{F}$  sigue siendo cerrada por unión. La primera de las dos pruebas es trivial con la codificación propuesta. En efecto, ya que el conjunto vacío corresponde al bit de menor peso, una familia contiene el conjunto vacío si y solamente si el identificador de esta familia es impar. La segunda prueba, más delicada, impone verificar que para cada elemento de  $\mathcal{F}$ , la unión de este elemento con  $E$  ya esté en  $\mathcal{F}$ . Esto puede ser hecho por el código C siguiente:

```
for ( i = 1 ; i < N ; i++ )
    if ( ( F & ( 1 << i ) ) && ! ( F & ( 1 << ( E | i ) ) ) )
        printf("F uni\'on E no es cerrado por uni\'on\n");
```

El ciclo *for* recorre el conjunto de los enteros  $i$  característicos de un conjunto. La primera parte de la prueba *if* ( $\mathcal{F} \& (1U \ll i)$ ) verifica si el conjunto que corresponde al entero  $i$  pertenece a la familia  $\mathcal{F}$ . Para eso se construye un vector constituido de ceros salvo un 1 en la posición  $i$ , con la expresión  $(1U \ll i)$  y se hace una *y* lógica con la familia  $\mathcal{F}$ . Esta prueba retorna naturalmente "cierto" si y solamente si el conjunto  $i$  pertenece a  $\mathcal{F}$ . La segunda parte de la prueba verifica si  $i \cup E$  pertenece a la familia  $\mathcal{F}$ . Como el identificador del conjunto  $E \cup i$  corresponde al valor entero de  $E|i$ , basta realizar una *y* lógica entre  $\mathcal{F}$  y un vector constituido de ceros salvo un 1 en la posición  $E|i$ .

## 2.2. Simetría y forma canónica

Una permutación  $\Phi$  sobre un conjunto finito  $U_n = \{0, \dots, n-1\}$  es una función biyectiva de  $U_n$  en  $U_n$ . Por comodidad  $\Phi$  es a menudo representado por la secuencia de sus imágenes  $\Phi(0), \Phi(1), \dots, \Phi(n-1)$ . Notamos  $Sym_n$  el conjunto de todas las permutaciones sobre un conjunto  $U_n$ . Para un conjunto  $E \subseteq U_n$  y  $\Phi \in Sym_n$  notamos  $\Phi(E)$  la imagen de  $E$  por  $\Phi$  definida por:  $\Phi(E) = \{\Phi(x) | x \in E\}$ . De la misma forma, para toda familia  $\mathcal{F} \subseteq 2^{U_n}$ , tenemos simplemente  $\Phi(\mathcal{F}) = \{\Phi(E) | E \in \mathcal{F}\}$ .

**Ejemplo:** sea la familia  $\mathcal{F} = \{\emptyset, \{2\}, \{1, 2\}, \{0, 1, 2\}\}$  y la permutación  $\Phi = 1, 2, 0$  entonces  $\Phi(\mathcal{F}) = \{\emptyset, \{0\}, \{0, 2\}, \{0, 1, 2\}\}$ .

Basándose en el concepto de permutación, podemos partir el conjunto de las familias sobre  $U_n$  en clases de equivalencia. Así, diremos que dos familias pertenecen a la misma clase, si son imágenes una de la otra por una permutación de  $Sym_n$ . Llamamos **forma canónica** la familia de

Permutación $\Phi$	$\Phi(\mathcal{F})$	Identificador
012	$\{\emptyset, \{2\}, \{12\}, \{012\}\}$	209
021	$\{\emptyset, \{1\}, \{12\}, \{012\}\}$	197
102	$\{\emptyset, \{2\}, \{02\}, \{012\}\}$	177
120	$\{\emptyset, \{0\}, \{02\}, \{012\}\}$	163
210	$\{\emptyset, \{0\}, \{01\}, \{012\}\}$	139
201	$\{\emptyset, \{1\}, \{01\}, \{012\}\}$	141

Cuadro 2: Para  $\mathcal{F} = \{\emptyset, \{2\}, \{1, 2\}, \{0, 1, 2\}\}$  y cada permutación  $\Phi$  de  $Sym_3$ , la imagen de  $\mathcal{F}$  por  $\Phi$  y su identificador.

referencia de cada clase. Hemos retenido una identificación de la forma canónica basada en las propiedades de nuestra codificación. Así, para una familia  $\mathcal{F} \subseteq 2^{U_n}$ , su forma canónica es definida como la imagen por la permutación de  $Sym_n$  que tiene el menor identificador.

**Ejemplo:** Consideremos la familia  $\mathcal{F} = \{\emptyset, \{2\}, \{1, 2\}, \{0, 1, 2\}\}$  sobre  $U_3$  así que el conjunto  $Sym_3$  conteniendo 6 permutaciones. Según la tabla 2, la forma canónica de  $\mathcal{F}$  es la familia  $\{\emptyset, \{0\}, \{01\}, \{012\}\}$  cuyo identificador vale 139.

**Proposición 1** Sean  $\mathcal{M}$  una familia de Moore sobre  $U_n$  y  $\Phi \in Sym_n$  una permutación, entonces  $\Phi(\mathcal{M})$  es una familia de Moore sobre  $U_n$ .

$n$	Familias de Moore no isomorfas	Tamaño promedio (clases)
1	2	1
2	5	1, 40
3	19	3, 21
4	184	13, 48
5	14664	94, 49
6	108 295 846	701, 54

Cuadro 3: Para cada valor de  $n$ , el número de familias de Moore no isomorfas y el tamaño promedio de las clases. El tamaño promedio de las clases aumenta exponencialmente con  $n$  (y viene aparentemente a  $n!$ ) lo que muestra que el conjunto de todas las familias de Moore sobre un universo contiene un gran número de objetos isomorfos.

La existencia de esa partición sobre el conjunto de las familias generales sobre  $U_n$  sigue pertinente respecto al conjunto de las familias de

Moore. Así mismo, según la propiedad anterior, la imagen de una familia de Moore por una permutación sigue siendo una familia de Moore. En consecuencia, es posible enumerar las familias de Moore enumerando el representante de cada clase de equivalencia y luego calculando sus imágenes por cada una de las permutaciones. De una forma más general, llamamos “enumeración de un conjunto de objetos combinatorios no isomorfos”, a la enumeración de un sólo representante por clase de equivalencia. Esta estrategia se basa a menudo sobre el hecho que una gran parte de la explosión combinatoria asociada al conjunto de los objetos estudiados se explica por la presencia de objetos isomorfos. La tabla 3 muestra que esta situación es particularmente verificada para el conjunto de las familias de Moore.

### 2.3. Familia máxima

Una familia de Moore  $\mathcal{M}$  sobre  $U_n$  puede ser descompuesta en 2 partes. La parte constituida de los conjuntos de  $\mathcal{M}$  que contienen el elemento  $\{n - 1\}$  (notada  $\mathcal{M}_{sup}$  por parte superior) y la parte complementaria (notada  $\mathcal{M}_{inf}$  por parte inferior). El elemento  $\emptyset$  es duplicado para estar presente en ambas partes. Naturalmente  $\mathcal{M} = \mathcal{M}_{sup} \cup \mathcal{M}_{inf}$ . Además,  $\mathcal{M}_{sup}$  y  $\mathcal{M}_{inf}$  son familias de Moore. El ejemplo de la figura 5 hace resaltar que para una parte superior dada, existen varias partes inferiores compatibles (es decir cuya reunión da una familia de Moore).

Para una familia superior  $\mathcal{M}_{sup}$  dada, existe una única **familia máxima** tal que todas las familias compatibles con  $\mathcal{M}_{sup}$  son subfamilias de la familia máxima. Más formalmente:

**Proposición 2** *Sea  $\mathcal{M}_{sup}$  una familia de Moore sobre  $U_n$  con  $n - 1 \in M$  para todo  $M \in \mathcal{M}_{sup} \setminus \{\emptyset\}$ . Existe entonces una única familia de Moore  $\mathcal{M}_{max}$  sobre  $U_{n-1}$ , compatible con  $\mathcal{M}_{sup}$  tal que todas las familias de Moore sobre  $U_n$  cuya restricción a sus elementos conteniendo  $n - 1$  coincide con  $\mathcal{M}_{sup}$  se escriba  $\mathcal{M}_{sup} \cup \mathcal{M}_{inf}$ , con  $\mathcal{M}_{inf} \subseteq \mathcal{M}_{max}$ .*

Como ejemplo, la familia máxima asociada con la familia  $\mathcal{M}_{sup}$ , dada en figura 5, es la familia compuesta de los elementos  $\{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$ . Se puede verificar que ambas familias compatibles dadas en figura 5 son efectivamente subfamilias de esta familia.

Recordamos que la codificación de una familia de Moore toma la forma de un vector de bits de tamaño  $2^n$  (para contar las familias de Moore sobre  $U_7$  usaremos entonces un vector de tamaño 128). Naturalmente los primeros  $2^{n-1}$  bits codifican por el conjunto de los conjuntos que contienen el elemento  $n - 1$  (es decir,  $\mathcal{M}_{sup}$ ), los últimos  $2^{n-1}$  bits codifican



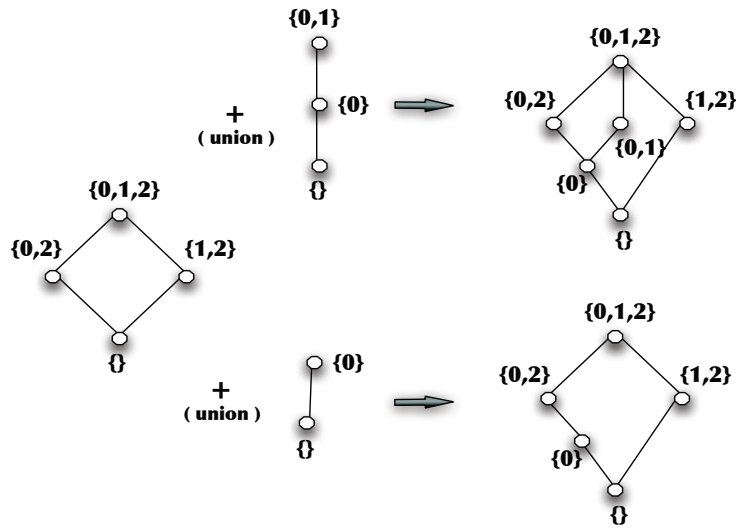


Figura 5: A la izquierda, una familia de Moore sobre  $U_3$  (cuyos elementos contienen 2) a la cual son adjuntas dos familias de Moore distintas sobre  $U_{n-1}$ . En ambos casos, la familia obtenida, a la derecha, es una familia de Moore.

entonces por  $\mathcal{M}_{inf}$ . Nuestra estrategia de conteo consiste entonces, en generar únicamente las partes superiores del vector, y para cada una de estas partes superiores en determinar cuál es la familia máxima compatible con ella (o sea, calcular la parte inferior del vector). El algoritmo 1 es un método de cálculo de esta familia única.

**Proposición 3** Sea  $\mathcal{M}_{sup}$  una familia de Moore sobre  $U_n$  (todos sus elementos contienen  $n$ ), el algoritmo 1 calcula la familia máxima  $\mathcal{M}_{max}$  compatible con  $\mathcal{M}_{sup}$  y retorna el vector representando la familia  $\mathcal{M}_{sup} \cup \mathcal{M}_{max}$ .

El interés del concepto de familia máxima es múltiple: primero, hemos visto que el conteo de las familias de Moore de  $U_7$  necesita el uso de vectores de 128 bits. Desafortunadamente, los enteros sobre 128 bits son poco prácticos para manipular directamente. El uso de las familias máximas permite descomponer el cálculo de las familias de Moore en dos partes distintas e independientes usando únicamente vectores de 64 bits. Además, eso permite también volver a usar cálculos hechos durante el conteo de las familias de Moore sobre  $U_{n-1}$  para contar las familias

---

**Algoritmo 1:** *familiaMaxima()*

---

**Datos:**  $V[2^n]$  representando la familia  $\mathcal{M}_{sup}$ **Resultado:**  $V[2^n]$  representando la familia  $\mathcal{M}_{sup} \cup \mathcal{M}_{max}$ **principio****para** ( $b = 2^n - 1$  a 0) **hacer** $V[b] \leftarrow 1$ ; **si**  $V[]$  no representa una familia cerrada por unión **entonces** $V[b] \leftarrow 0$ ;**fin****fin****devolver**  $V[]$ ;**fin**

---

de Moore sobre  $U_n$ . Al fin, permite resaltar una cota natural sobre el número de familias de Moore sobre  $U_n$  en función de este número sobre  $U_{n-1}$  (véase proposición 4).

**Proposición 4** Sean  $M_n$  y  $M_{n+1}$  los conjuntos de todas las familias de Moore respectivamente sobre  $U_n$  y  $U_{n+1}$ . Entonces tenemos  $|M_{n+1}| \leq 2 * |M_n|^2$ .

La sección siguiente trata de la realización de estos conceptos en un entorno algorítmico.

### 3. Algoritmo

Esta sección describe tres algoritmos distintos de conteo de las familias de Moore sobre  $U_n$ . El primero es un algoritmo ingenuo y recursivo que recorre un árbol representando el conjunto de las familias de Moore. El segundo introduce el concepto de simetría y almacena en una tabla hash el representante único de cada clase. Una variable “coeficiente”, asociada a cada clase, almacena el tamaño de la clase. Al fin, el tercer algoritmo integra a la estrategia anterior el concepto de familia máxima. Hay que notar que sólo este último algoritmo admite tiempos de cálculo razonables para  $n = 7$ .

#### 3.1. Algoritmo ingenuo

El algoritmo 2 realiza un recorrido recursivo del conjunto de las familias de Moore sobre  $U_n$ . El soporte de este recorrido es un árbol cuyas hojas son las familias de Moore y cuyas aristas registran la presencia o

la ausencia de uno u otro conjunto en la familia. El árbol de las familias de Moore sobre  $U_2$  es dado en la figura 6. Así, para cada nodo, el algoritmo genera un nuevo conjunto y determina si puede ser agregado a la familia actual. Si la respuesta es "no", el proceso continua sobre la rama de la izquierda con la misma familia. En el caso contrario, el proceso es reiniciado sobre la rama de la derecha con una familia que integra el nuevo elemento. Cuando se llega a una hoja, una variable global ( $|M_n|$ ), que cuenta el número total de familias, es incrementada.

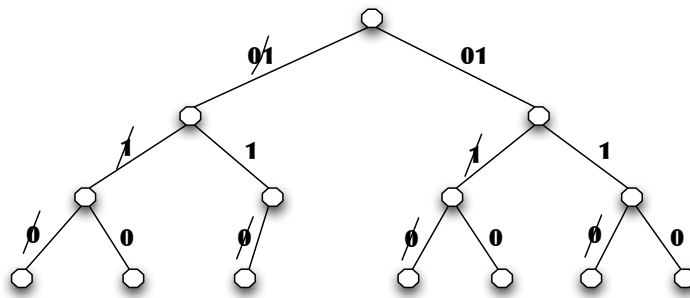


Figura 6: El árbol binario que corresponde a la exploración del algoritmo 2.

---

**Algoritmo 2:**  $Moore_1()$

---

**Datos:**  $V[]$ : vector de bits;  $k$ : entero

**principio**

**si**  $k = 0$  **entonces**

$|M_n| \leftarrow |M_n| + 1;$

**fin**

**sino**

$Moore_1(V[], k-1); V[k] \leftarrow 1;$  **si**  $V[]$  *representa una familia cerrada por unión* **entonces**

$Moore_1(V[], k-1)$

**fin**

$V[k] \leftarrow 0;$

**fin**

**fin**

---

Técnicamente, la familia corriente es almacenada en el vector  $V[]$  (inicializado a 0) que será actualizado desde la izquierda hacia la derecha.  $k$  corresponde a la posición del bit de  $V[]$ , es decir al conjunto que es

susceptible de ser insertado en la familia.  $k$ , inicializado a  $2^n - 1$  decrece naturalmente hasta 0, que corresponde a una hoja del árbol. De esta manera, los conjuntos son considerados en un orden total cuyo primer elemento es el conjunto  $\{0, 1, \dots, n - 1\}$  y cuyo último elemento es el conjunto vacío.

### 3.2. Algoritmo con simetrías

La idea del algoritmo presentado aquí integra el concepto de simetrías y almacena en una tabla hash las formas canónicas, asociando un *coeficiente* a cada una de ellas. La inserción de una forma canónica acoplada a un coeficiente  $c$  en la tabla consiste en crear una nueva entrada si la forma no es conocida o en sumar  $c$  al coeficiente de la forma ya presente.

Para calcular la forma canónica de la familia de Moore  $\mathcal{M}$ , la función  $formaCanonica(\mathcal{M})$  genera las imágenes de  $\mathcal{M}$  por todas las permutaciones de  $Sim_{U_n}$  y retorna aquella con el menor identificador. Los pequeños valores de  $n$  considerados (máximo 7) hacen que el tiempo de cálculo sea razonable con una complejidad  $O(n!)$ . Notemos al fin que el problema del cálculo de la forma canónica se reduce al problema del isomorfismo de grafos.

Nuestro método consta de dos algoritmos. El primero  $llenaVector()$  (véase el algoritmo 3) permite asignar bits al vector  $V[]$  entre las posiciones  $k$  y  $m$  de tal forma que  $V[]$  sea una familia cerrada por unión. Se trata de una adaptación del algoritmo  $Moore_1$  para la cual el último bit (correspondiendo a una hoja) se puede escoger. El segundo algoritmo (véase el algoritmo 4) utiliza a cada etapa las informaciones almacenadas en la tabla hash para construir eficientemente nuevos vectores susceptibles de corresponder a familias de Moore. Al final del cálculo, basta sumar los coeficientes de cada forma canónica presente en la última tabla para conocer el número total de familias.

Vemos que el coeficiente  $c$  asociado a un vector  $V[]$  durante su inserción en la tabla hash es el coeficiente asociado al vector que fue utilizado durante la llamada al algoritmo  $llenaVector()$  en la función  $Moore_2$ .

La tabla 4 resume para cada valor de  $n$  el tamaño de la tabla hash conteniendo las familias generadas. El tamaño aparece antes y después de la reducción. La columna a la derecha corresponde a los números de Moore equivalentes mediante un isomorfismo. Recordamos que la tabla hash antes de la reducción nunca es construida porque la reducción se hace a lo largo del proceso mediante la función  $formaCanonica()$ .

---

**Algoritmo 3:** *llenaVector()*

---

**Datos:**  $V[]$  : vector de bits;  $k, m$  : enteros;**principio****si**  $k = m - 1$  **entonces** $miTablaHash.agregar(formaCanonica(V[]));$ **fin****sino** $llenaVector(V[], k - 1); V[k] \leftarrow 1;$  **si**  $V[]$  *representa una familia cerrada por unión* **entonces** $llenaVector(V[], k - 1);$ **fin** $V[k] \leftarrow 0;$ **fin****fin**

---

---

**Algoritmo 4:** *moore2()*

---

**Datos:**  $n$  : entero;**Resultado:** entero, número de familias de Moore sobre  $U_n$ ;**principio** $H'.agregar(V[] = \langle 0, 0, \dots, 0, 0, 1 \rangle); H \leftarrow \emptyset; i \leftarrow 1;$  **mien-****tras**  $i \leq n$  **hacer****para**  $V[] \in H'$  **hacer** $llenaVector(V[], 2^i - 1, 2^{i-1});$ **fin** $i ++; H' \leftarrow H;$ **fin****devolver**  $\Sigma_{V[] \in H} V[].coeficiente;$ **fin**

---

$n$	$ H $ antes de la reducción	$ H $ después de la reducción
2	7	5
3	46	19
4	916	184
5	140 463	14 664
6	770 413 085	108 295 846

Cuadro 4: Los tamaños de la tabla hash conteniendo las familias de Moore para los distintos valores de  $n$  antes y después de la reducción.

### 3.3. Algoritmo usando las familias máximas

El algoritmo presentado aquí integra el concepto de familia máxima (véase sección anterior) y el concepto de simetría que acabamos de desarrollar. Primero, una adaptación del algoritmo anterior es usada para generar los vectores que representan las familias  $\mathcal{M}_{sup}$  equivalentes mediante un isomorfismo (el número de familias isomorfas a cada una de estas familias es almacenado en el coeficiente  $c$ ). Segundo, se determinan para cada una de las familias  $\mathcal{M}_{sup}$ , la familia máxima  $\mathcal{M}_{inf}$  asociada. Recordamos que una familia  $\mathcal{M}_{inf}$  es una familia de Moore sobre  $U_{n-1}$  y que su peso  $p$  corresponde al número de familias de Moore incluidas en esta familia. El conteo de las familias de Moore sobre  $U_n$  corresponde entonces a la suma de los productos de los pesos de cada familia y de sus coeficientes.

La función *familiaMaxima()* toma como entrada un vector  $V[]$  de  $2^n$  bits que representa una familia de Moore, sobre  $U_n$  tal que todos sus elementos contengan el elemento  $n - 1$ . En consecuencia, los bits de  $V[]$  ubicados entre 1 y  $2^{n-1} - 1$  son todos inicializados a 0. Retorna un vector de  $2^{n-1}$  bits representando la familia de Moore sobre  $U_{n-1}$  que corresponde a la familia máxima asociada a la familia de Moore considerada.

El algoritmo *parteIzquierda()* (vease el algoritmo 5) genera los vectores equivalentes mediante un isomorfismo, que corresponden a las familias de Moore sobre  $U_n$  considerando solamente los elementos de  $2^{U_n} \setminus 2^{U_{n-1}} \setminus \{n - 1\}$ . Es una adaptación del algoritmo *Moore2* dado anteriormente, considerando únicamente los bits entre  $2^n - 1$  y  $2^{n-1} + 1$ .

El algoritmo *Moore3()* (vease el algoritmo 6) empieza generando las partes izquierdas de los vectores usando el algoritmo *parteIzquierda()*. A cada parte izquierda se asocian dos familias de Moore: una que contiene el elemento  $n - 1$  y una otra que no lo contiene. El algoritmo asocia



---

**Algoritmo 7:** *calculaPeso()*

---

**Datos:**  $V[]$  : vector,  $k$  : entero;**Resultado:**  $p$ ;**principio**  **si**  $k = 2^n$  **entonces**     $p \leftarrow p + 1$ ;  **fin****sino**   $calculaPeso(V[], k + 1)$ ;  $V[k] \leftarrow 0$ ; **si**  $V[]$  *representa una familia cerrada por unión* **entonces**     $calculaPeso(V[], k + 1)$   **fin****fin****fin**

---

## 4. Aspectos técnicos

### 4.1. Implementación

El programa fue escrito en C, compilado con GCC versión 4 para Linux 64 bits. Hemos usado la biblioteca de programación paralela OpenMP. El cálculo principal duró 11 horas sobre un Core2 Q9300 a 2,5 GHz. La verificación duró 9 horas sobre un doble Xeon E5320 a 1,86 GHz (Departamento de Matemáticas, UNAL). El programa corrió también sobre un cuádruple Opteron 8356 a 2,3 GHz (Universidad de Clermont-Ferrand, Francia).

### 4.2. Fiabilidad

La cuestión de la fiabilidad es central para cualquier resultado implicando un gran número de cálculos computacionales. En el caso presente, los autores quieren resaltar los puntos siguientes:

- El resultado es compatible con las estimaciones anteriores de uno de los autores (entre  $1,2 \cdot 10^{19}$  y  $1,7 \cdot 10^{19}$ ) basadas sobre una extrapolación de los datos para  $n \leq 6$ .
- El mismo programa produce resultados correctos para  $n$  entre 1 y 6 (número de familias de Moore y de familias reducidas por isomorfismo).
- No hay código específico a  $n = 7$  y no hay barrera particular entre  $n = 6$  y  $n = 7$  (tamaño de los enteros usados por ejemplo).



- El código fue ejecutado varias veces sobre arquitecturas distintas (Intel y AMD).
- El programa fue compilado con tres 3 compiladores conocidos como estables (GCC v. 4.1, 4.2 y 4.3) con diversas opciones, sobre sistemas operativos distintos (Ubuntu y CentOS).
- El resultado no sufrió alteraciones debido a modificaciones en el tamaño disponible para las tablas hash. Una modificación del tamaño de las tablas provoca una modificación completa del orden de almacenamiento de los datos que a su turno provoca elecciones distintas a la iteración siguiente.
- Al reemplazar la forma canónica basada en el menor representante de la clase de equivalencia por una forma canónica basada en el mayor, el resultado no se altera.

## 5. Conclusión

El problema de la enumeración de las familias de Moore al rango  $n$  es un problema difícil para el cual no existe ninguna fórmula conocida. Tampoco fue demostrado que semejante fórmula no existiera. Muchos problemas combinatorios están en este caso. Se pueden citar por ejemplo el número de funciones booleanas monótonas, conocido bajo el nombre de problema de Dedekind. Un enfoque frecuente para acercarse a tales fórmulas consiste en contar el número de objetos para los primeros valores de  $n$  mediante enumeración directa. Se pueden encontrar tales secuencias de enteros sobre la conocida On-Line Encyclopedia of Integer Sequences <sup>1</sup>. Nuestro trabajo tenía entonces un doble objetivo: no solamente enriquecer la secuencia ya conocida para el número de familias de Moore, sino también poner en evidencia nuevas propiedades del conjunto de las familias de Moore.

En este artículo, hemos mostrado que podíamos contar los objetos considerados sin enumerarlos todos. Así, existe una secuencia de objetos más corta (las familias de Moore sobre  $U_n$  reducidas por isomorfismo cuyos conjuntos contienen todos  $n - 1$ ) a partir de la cual es posible determinar el número total. Más precisamente, para cada objeto de la secuencia, “basta” calcular su familia máxima asociada (una familia de Moore sobre  $U_{n-1}$ ) y calcular su peso (es decir el número de familias de Moore incluidas). Este cálculo de enumeración es realizado una sola vez

---

<sup>1</sup><http://www.research.att.com/~njas/sequences/>

incluso si la familia máxima se encuentra varias veces. El resultado es almacenado en una tabla hash cuya entrada corresponde al identificador de la familia máxima. Podemos también afirmar que cada familia de Moore sobre  $U_{n-1}$  aparece por lo menos una vez como familia máxima.

Ser capaces de predecir cuántas veces cada familia sobre  $U_{n-1}$  aparece como familia máxima de una familia sobre  $U_n$ , nos acercaría mucho de una fórmula general, si es que existe una.

## 6. Anexo

**Proposición 1** Sean  $\mathcal{M}$  una familia de Moore sobre  $U_n$  y  $\Phi \in \text{Sym}_n$  una permutación, entonces  $\Phi(\mathcal{M})$  es una familia de Moore sobre  $U_n$ .

*Prueba:* Mostremos que  $\Phi(\mathcal{M})$  es una familia de Moore.

- $\emptyset \in \mathcal{M}$ ,  $\Phi(\emptyset) = \emptyset$  entonces  $\emptyset \in \Phi(\mathcal{M})$ .
- Sean  $m_1$  y  $m_2 \in \mathcal{M}$  y  $\Phi(m_1)$  y  $\Phi(m_2) \in \Phi(\mathcal{M})$ . Tenemos  $\Phi(m_1) \cup \Phi(m_2) = \Phi(m_1 \cup m_2)$  por definición de  $\Phi()$ . Ya que  $\mathcal{M}$  es cerrada por unión  $m_1 \cup m_2 \in \mathcal{M}$ . Y entonces  $\Phi(m_1 \cup m_2) \in \Phi(\mathcal{M})$ .

Asimismo  $\forall \Phi(m_1)$  y  $\Phi(m_2) \in \Phi(\mathcal{M})$ ,  $\Phi(m_1) \cup \Phi(m_2) \in \Phi(\mathcal{M})$ .

$\Phi(\mathcal{M})$  contiene al conjunto vacío y es cerrada por uniones,  $\Phi(\mathcal{M})$  es una familia de Moore.

**Proposición 2** Sea  $\mathcal{M}_{sup}$  una familia de Moore sobre  $U_n$  con  $n - 1 \in M$  para todo  $M \in \mathcal{M}_{sup} \setminus \{\emptyset\}$ . Entonces existe una única familia de Moore  $\mathcal{M}_{max}$  sobre  $U_{n-1}$  tal que todas las familias de Moore sobre  $U_n$  cuya restricción a sus elementos conteniendo  $n - 1$  coincida con  $\mathcal{M}_{sup}$  se escribiesen  $\mathcal{M}_{sup} \cup \mathcal{M}_{inf}$ , con  $\mathcal{M}_{inf} \subseteq \mathcal{M}_{max}$ .

*Prueba:* Sea  $\mathcal{M}$  una familia de Moore sobre  $U_n$  con  $\mathcal{M} = \mathcal{M}_{sup} \cup \mathcal{M}_{inf}$ .

A) Mostremos que para toda familia de Moore  $\mathcal{M}'_{inf}$  sobre  $U_{n-1}$  con  $\mathcal{M}'_{inf} \subseteq \mathcal{M}_{inf}$ ,  $\mathcal{M}_{sup} \cup \mathcal{M}'_{inf}$  es una familia de Moore.

- $\emptyset \in \mathcal{M}'_{inf}$ , entonces  $\emptyset \in \mathcal{M}_{sup} \cup \mathcal{M}'_{inf}$ ;
- sean  $m_1$  y  $m_2 \in \mathcal{M}_{sup} \cup \mathcal{M}'_{inf}$ ; 3 casos ocurren:
  1.  $m_1$  y  $m_2 \in \mathcal{M}_{sup}$ , como  $\mathcal{M}_{sup}$  es una familia de Moore  $m_1 \cup m_2 \in \mathcal{M}_{sup}$  y entonces  $m_1 \cup m_2 \in \mathcal{M}_{sup} \cup \mathcal{M}'_{inf}$ ;

2.  $m_1$  y  $m_2 \in \mathcal{M}'_{inf}$ , como  $\mathcal{M}'_{inf}$  es una familia  $m_1 \cup m_2 \in \mathcal{M}'_{inf}$  y entonces  $m_1 \cup m_2 \in \mathcal{M}_{sup} \cup \mathcal{M}'_{inf}$ ;
3.  $m_1 \in \mathcal{M}_{sup}$  y  $m_2 \in \mathcal{M}'_{inf}$ . Ya que  $\mathcal{M}'_{inf} \subseteq \mathcal{M}_{inf}$  y que para todo  $m \in \mathcal{M}_{inf}$ ,  $m_1 \cup m_2 \in \mathcal{M}_{sup}$  (porque  $m \cup m_1$  contiene  $n$ ) entonces  $m_1 \cup m_2 \in \mathcal{M}_{sup}$ . Y así  $m_1 \cup m_2 \in \mathcal{M}_{sup} \cup \mathcal{M}'_{inf}$ ;

B) Mostremos que para toda familia de Moore  $\mathcal{M}'_{inf}$  y  $\mathcal{M}''_{inf}$  sobre  $U_{n-1}$  tal que  $\mathcal{M}_{sup} \cup \mathcal{M}'_{inf}$  y  $\mathcal{M}_{sup} \cup \mathcal{M}''_{inf}$  sean familias de Moore entonces  $C(\mathcal{M}'_{inf} \cup \mathcal{M}''_{inf}) \cup \mathcal{M}_{sup}$  es una familia de Moore donde  $C$  es el operador de clausura por la operación de unión.

- $\emptyset \in \mathcal{M}'_{inf}$ , entonces  $\emptyset \in C(\mathcal{M}'_{inf} \cup \mathcal{M}''_{inf}) \cup \mathcal{M}_{sup}$ ;
- Sean  $m$  y  $m' \in C(\mathcal{M}'_{inf} \cup \mathcal{M}''_{inf}) \cup \mathcal{M}_{sup}$ ; el único caso difícil es el caso donde  $m \in \mathcal{M}_{sup}$  y  $m' \in C(\mathcal{M}'_{inf} \cup \mathcal{M}''_{inf}) \setminus \mathcal{M}'_{inf} \setminus \mathcal{M}''_{inf}$ . Como  $\mathcal{M}'_{inf}$  y  $\mathcal{M}''_{inf}$  son familias de Moore entonces existe  $m_1$  en  $\mathcal{M}'_{inf}$  y  $m_2$  en  $\mathcal{M}''_{inf}$  tales que  $m' = m_1 \cup m_2$  y
  1. si  $m_1 \in \mathcal{M}'_{inf}$  entonces  $\forall m \in \mathcal{M}_{sup}$  tenemos  $m \cup m_1 \in \mathcal{M}_{sup}$ ;
  2. si  $m_2 \in \mathcal{M}''_{inf}$  entonces  $\forall m \in \mathcal{M}_{sup}$  tenemos  $m \cup m_2 \in \mathcal{M}_{sup}$ ;

Y como  $\mathcal{M}_{sup}$  es una familia de Moore  $m_1 \cup m_2 \cup m = m' \cup m$  pertenece a  $\mathcal{M}_{sup}$

Así podemos concluir que para cada pareja de elementos  $m$  y  $m'$  en  $C(\mathcal{M}'_{inf} \cup \mathcal{M}''_{inf}) \cup \mathcal{M}_{sup}$ ,  $m \cup m'$  pertenece a  $\mathcal{M}_{sup}$ . Entonces  $C(\mathcal{M}'_{inf} \cup \mathcal{M}''_{inf}) \cup \mathcal{M}_{sup}$  es una familia de Moore.

Se deduce de B) que el conjunto de las familias compatibles con una familia  $\mathcal{M}_{sup}$  dada, admite un sólo elemento máximo llamado  $\mathcal{M}_{max}$  tal que para toda familia  $\mathcal{M}_{inf}$  incluida en o igual a  $\mathcal{M}_{max}$ , la reunión de  $\mathcal{M}_{sup}$  y  $\mathcal{M}_{inf}$  sea una familia de Moore.

**Proposición 3** Sean  $\mathbb{M}_{n-1}$  y  $\mathbb{M}_n$  los conjuntos de todas las familias de Moore respectivamente sobre  $U_{n-1}$  y  $U_n$ . Entonces tenemos  $|\mathbb{M}_n| \leq 2 * |\mathbb{M}_{n-1}|^2$ .

*Prueba:* Cada familia  $\mathcal{M} \in \mathbb{M}_{n-1}$  va a dar dos partes superiores  $\mathcal{M}_{sup_1} = \{M \cup n \mid M \in \mathcal{M}\} \cup \emptyset$  y  $\mathcal{M}_{sup_2} = \{M \cup n \mid m \in \mathcal{M}\} \cup \emptyset \setminus n$ . En consecuencia, el conjunto  $\mathbb{M}_{sup_n}$  de todas las familias  $\mathcal{M}_{sup}$  tiene cardinal  $2 * |\mathbb{M}_{n-1}|$ . En el peor de los casos la familia máxima asociada a cada familia  $\mathcal{M}_{sup} \in \mathbb{M}_{sup_n}$  es la familia de Moore  $2^{U_{n-1}}$  de peso  $|\mathbb{M}_{n-1}|$ . Entonces  $|\mathbb{M}_n| \leq 2 * |\mathbb{M}_{n-1}|^2$ .

**Proposición 4** Sean  $\mathbb{M}_n$  y  $\mathbb{M}_{n+1}$  los conjuntos de todas las familias de Moore respectivamente sobre  $U_n$  y  $U_{n+1}$ . Entonces tenemos  $|\mathbb{M}_{n+1}| \leq 2 * |\mathbb{M}_n|^2$ .

*Prueba:* Cada familia  $\mathcal{M} \in \mathbb{M}_n$  va a dar dos partes superiores  $\mathcal{M}_{sup_1} = \{M \cup n \mid M \in \mathcal{M}\} \cup \emptyset$  y  $\mathcal{M}_{sup_2} = \{M \cup n \mid m \in \mathcal{M}\} \cup \emptyset \setminus n$ . En consecuencia, el conjunto  $\mathbb{M}_{sup_{n+1}}$  de todas las familias  $\mathcal{M}_{sup}$  tiene cardinal  $2 * |\mathbb{M}_n|$ . En el peor de los casos la familia máxima asociada a cada familia  $\mathcal{M}_{sup} \in \mathbb{M}_{sup_{n+1}}$  es la familia de Moore  $2^{U_n}$  de peso  $|\mathbb{M}_n|$ . Entonces  $|\mathbb{M}_{n+1}| \leq 2 * |\mathbb{M}_n|^2$ .

## Referencias

- [1] G. Birkhoff, *Lattice Theory* (American Mathematical Society, 1967).
- [2] G. Burosh, J. Demetrovics, G. O. H. Katona, D. J. Kleitman and A. A. Sapozhenko, *On the number of databases and closure operations*, Theor. Comp. Sci. **78**, 377–381 (1991).
- [3] N. Caspard and B. Monjardet, *The lattices of closure systems, closure operators, and implicational systems on a finite set: a survey*, Disc. Appl. Math. **127**, 241–269 (2003).
- [4] P. M. Cohn, *Universal Algebra* (Harper and Row, New York, 1965).
- [5] B. A. Davey and H. A. Priestley, *Introduction to lattices and orders* (Cambridge University Press, 1991).
- [6] J. Demetrovics, L. Libkin and I. B. Muchnik, *Functional dependencies in relational databases: A lattice point of view*, Disc. Appl. Math. **40**, 155–185 (1992).
- [7] V. Duquenne, *Latticial structure in data analysis*, Theor. Comp. Sci. **217**, 407–436 (1999).
- [8] B. Ganter and R. Wille, *Formal concept analysis*, in Mathematical foundation (Springer, Berlin–Heidelberg–New York, 1999).
- [9] M. Habib and L. Nourine, *The number of Moore families on  $n=6$* , Disc. Math. **294**, 291–296 (2005).
- [10] A. Higuchi, *Note: Lattices of closure operators*, Disc. Math. **179**, 267–272 (1998).

- [11] R. Medina and L. Nourine, *Algorithme efficace de génération des idéaux d'un ensemble ordonné*, *Compte rendu de l'Académie des sciences*, Paris **319**, 1115–1120 (1994).
- [12] E. H. Moore, *Introduction to a form of general analysis* (Yale University Press, New Haven, 1910).