Un paquete para cálculos en reticulados

Cecilia R. Cimadamore y Sebastián J. Ferraro
Departamento de Matemática
Universidad Nacional del Sur, Av. Alem 1253
8000 Bahía Blanca
{crcima,sferraro}@criba.edu.ar

Resumen

En este trabajo presentamos un paquete que hemos desarrollado para manejar y visualizar reticulados y álgebras de Łukasiewicz finitos. Este provee al investigador matemático herramientas para realizar en forma sencilla cálculos que normalmente conllevan un arduo trabajo. El paquete ha sido probado con numerosos ejemplos por miembros de un grupo de investigación en lógica algebraica de nuestra universidad. Su diseño orientado a objetos prevé su expansión futura.

Palabras clave: álgebra computacional, informática aplicada a matemática, lógica algebraica, reticulados distributivos.

1. Introducción

Los investigadores en el área de la lógica algebraica han realizado tradicionalmente a mano los cálculos en reticulados con estructura matemática adicional. Para ejemplos relativamente pequeños, éstos se hacen extremadamente engorrosos, requieren tiempos de trabajo prolongados y la probabilidad de cometer errores aumenta. Existe una necesidad concreta en los grupos de trabajo de generar ejemplos para verificar hipótesis. Para ciertos ejemplos de mediana complejidad, la forma de trabajo usual requiere tiempos prohibitivamente largos.

La utilización de la computadora surge naturalmente como una solución a este problema. Dada la falta de programas adecuados para el manejo de estas estructuras algebraicas, hemos diseñado e implementado un paquete para Matlab con el fin de facilitar la labor del investigador.

2. Reticulados y Algebras de Łukasiewicz

Un *reticulado* L es un poset (conjunto parcialmente ordenado) donde todo par de elementos posee ínfimo y supremo, operaciones denotadas por $\wedge$ y $\vee$. En este trabajo consideraremos únicamente reticulados finitos que posean primer y último elemento y que sean distributivos, esto es, $x\wedge(y\vee z) = (x\wedge y)\vee(x\wedge z)$ para todo $x,y,z\in L$.

Se define la *sección superior* de un subconjunto S en un poset dado P como el conjunto $[S) = \{x\in P | x\geq s,\ \text{para algún } s\in S\}$. Se dice que S es *creciente* si $[S) = S$. Los subconjuntos crecientes de P ordenados por la relación de inclusión forman un reticulado que notaremos $D(P)$. Este reticulado es de capital importancia en la teoría general de reticulados distributivos. (ver [7, 8]).

Un operador $C\colon L\to L$ se dice *de clausura* si verifica las siguientes propiedades para todo $x,y\in L$:

(C1) $C0 = 0$,

(C2) $x \leq Cx$,

(C3) $C(x \wedge y) = Cx \wedge Cy$,

(C4) $CCx = Cx$.

Los elementos $x \in L$ tales que $Cx = x$ se dicen *cerrados*, y el conjunto de elementos cerrados de L se denota por $C(L)$. Es posible reconstruir el operador C unívocamente a partir de $C(L)$ (ver por ejemplo [1]). Dado un subconjunto S de un reticulado finito L, existe un operador de clausura C tal que $S = C(L)$ si y sólo si

(c1) S es un *subreticulado* de L, es decir, contiene al primer y al último elemento de L y es cerrado por $\wedge$ y $\vee$

(ver por ejemplo [1]). El operador C es la aplicación que a cada $x \in L$ le asigna el primer elemento de $\{y \in S | x \leq y\}$, cuya existencia puede ser probada usando (c1) y la condición de finitud.

La noción de álgebras de Łukasiewicz de orden n, o n-valentes, fue introducida por Gr. C. Moisil [4]. Presentaremos aquí las definiciones básicas, pudiendo el lector encontrar más información en [5, 2, 6, 3, 9, 10].

Un *álgebra de Łukasiewicz n-valente L*, n entero mayor o igual a 2, es un reticulado distributivo con un operador unario $\sim$ de negación de de Morgan y $n-1$ operadores unarios $\varphi_1, \ldots, \varphi_{n-1}$ satisfaciendo una serie de identidades.(referenicias) Todos estos operadores toman valores en L.

El ejemplo más importante de un álgebra de Łukasiewicz de orden n es la cadena de n elementos C_n cuyos elementos son 0, $\frac{1}{n-1}$, $\frac{2}{n-1}$, $\ldots$, 1, con el orden usual. La negación está dada por $\sim \frac{j}{n-1} = 1 - \frac{j}{n-1}$, y los operadores φ_i son

$$\varphi_i\left(\frac{j}{n-1}\right) = \begin{cases} 0, & \text{si } i+j < n, \\ 1, & \text{si } i+j \geq n. \end{cases}$$

Las subálgebras de C_n son cadenas en cuanto a su estructura de posets, y heredan los operadores y el orden de C_n. Es sabido que toda álgebra de Łukasiewicz finita de orden n es producto directo de subálgebras de C_n (ver [3]).

Un álgebra de Łukasiewicz se dice *de clausura* si está provista de un operador de clausura C que verifica la propiedad adicional

(C5) $C\varphi_i x = \varphi_i Cx, 1 \leq i \leq n-1, x \in L$.

Es un problema abierto encontrar una condición adicional sobre S que asegure que C satisfaga además (C5). El enfoque actual es construir C a partir de S y comprobar la validez de (C5).

En un álgebra de Łukasiewicz de clausura es importante calcular los elementos x que verifican $\sim C \sim x = x$, denominados *abiertos*.

3. El paquete Lattice

En vista de la necesidad de generar ejemplos de prueba y de realizar cálculos, hemos diseñado e implementado un paquete de software que permite trabajar con posets, reticulados y álgebras de Łukasiewicz. Consiste en un toolbox para Matlab, que introduce clases `poset`, `lattice` y `luka`, y métodos adecuados para entrada, manejo, cálculo y visualización de datos. Se eligió Matlab como lenguaje de implementación debido a que el usuario al que apunta este desarrollo está familiarizado con su interfaz, y dadas sus adecuadas capacidades para OOP.

3.1. Principales características del paquete

El paquete permite al usuario definir objetos en el entorno Matlab. Un objeto `poset` representa un conjunto parcialmente ordenado finito, y contiene la relación de orden y etiquetas de texto para los elementos. Pueden construirse tanto cadenas de longitud arbitraria como posets con relaciones especificadas por el usuario mediante fórmulas numéricas. También se puede construir el producto directo, la suma ordinal y la unión disjunta de posets.

Dados dos elementos de un objeto `poset` (identificados por sus etiquetas), éste puede informar si se encuentran en relación y calcular su supremo e ínfimo. Dado un subconjunto S, determina $\bigvee_{x\in S} x$ y $\bigwedge_{x\in S} x$. Calcula además la sección superior de un elemento x, intersectada opcionalmente con un subconjunto S dado como argumento, es decir, $\{y \in S | x \leq y\}$ (análogamente para la sección inferior).

Esta clase posee una implementación de un algoritmo propio para la creación de diagramas de Hasse, lo que resulta una herramienta de mucha utilidad en la visualización. Opcionalmente puede resaltar un subconjunto de elementos dado.

La clase `lattice` está derivada de `poset`, y contiene tablas para los operadores de ínfimo y supremo a fin de agilizar cálculos posteriores. Puede construirse un `lattice` a partir de un poset P dado de dos formas distintas. La primera es copiar el `poset` como está, comprobando opcionalmente que efectivamente sea un reticulado (distributivo). La segunda es crear el reticulado L tal que $L = D(P)$.

Dado un subconjunto S de un reticulado L, la clase posee métodos para determinar si se trata de un subreticulado. En este caso puede construir el operador de clausura C tal que $C(L) = S$. También puede comprobar si un operador dado por el usuario es de clausura, y determinar los elementos cerrados.

La clase `luka` está derivada de `lattice` e incorpora las estructuras propias de un álgebra de Łukasiewicz. El usuario puede construir objetos `luka` como cadenas C_n, subálgebras de C_n y productos directos de éstas.

Dado un operador de clausura sobre el reticulado distributivo subyacente, verifica la validez de (C5), comprobando así que es un operador de clausura de Łukasiewicz, pudiéndose calcular el conjunto de elementos abiertos.

Todas estas clases poseen métodos para presentar los operadores de una forma útil para el usuario. El programa genera tablas en formato LaTeX y csv, permitiendo su inclusión en documentos o su manipulación en planillas de cálculo.

3.2. Ejemplos

La siguiente porción de código define un poset P usando la relación “divide” entre naturales y calcula el reticulado D del cual es dual. Se grafican también los diagramas de Hasse respectivos (Figura 1).

```
P=poset([1 2 3 9 18],'xRy <=> mod(y,x)==0');
D=lattice(P,'dual');
hasse(P);
hasse(D);
```

En el siguiente ejemplo se muestra cómo se define el álgebra de Łukasiewicz $L = C_2 \times C_2 \times C_3$ y se calcula el operador de clausura sobre el reticulado subyacente, asociado a un cierto subconjunto de posibles cerrados. En caso de ser además un operador de clausura de Łukasiewicz, calcula los abiertos. También se grafican dos diagramas de Hasse de L, resaltando los elementos abiertos y los cerrados (Figura 2). Finalmente, se crean tablas para el operador de clausura y para los operadores φ_i.

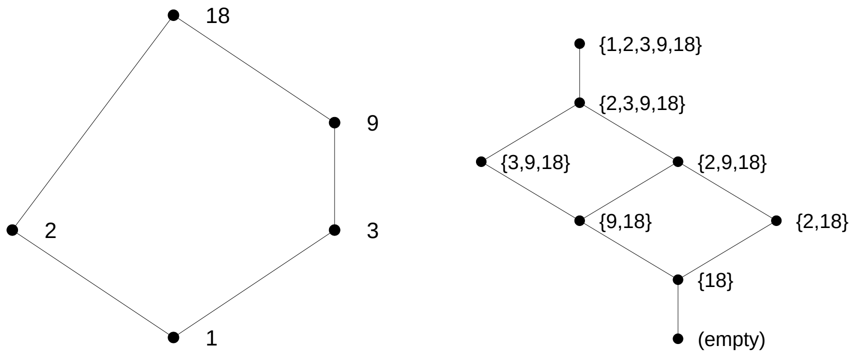

Figura 1: Diagramas de Hasse de P y D generados por la función `hasse`

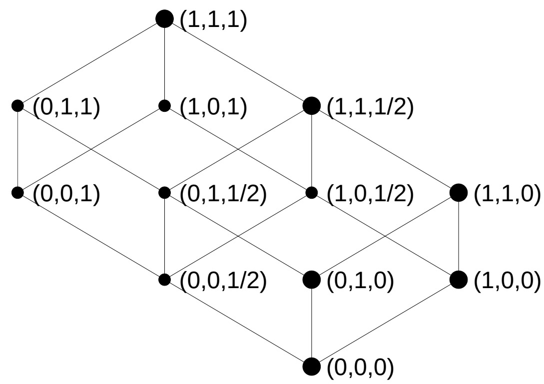

Figura 2: Diagrama de Hasse de L con los elementos abiertos resaltados

```
A=luka(3);
B=subcadenaluka(A,{'0','1'});
L=B*B*A;
S={'(0,0,0)','(0,0,1/2)','(0,0,1)','(1,0,1)','(0,1,1)','(1,1,1)'};
hasse(L,S);   % resalta los elementos de S en el Hasse de L
C=opclausura(L,S);   % crea, si es posible, el operador
                     % de clausura C con C(L)=S

if esclauluka(L,C)
    fprintf('Es de clausura de Lukasiewicz\n');
    Q=abiertos(L,C);
    hasse(L,Q);
end

creartablaop(L,C,'clausura.csv');   % en formato csv (default)
creartablaphi(L,'phi.tex','latex');
```

La siguiente tabla se generó insertando el archivo `phi.tex` en este punto del documento y quitando las cuatro últimas columnas para mayor claridad.

	(0,0,0)	(0,0,1/2)	(0,0,1)	(0,1,0)	(0,1,1/2)	(0,1,1)	(1,0,0)	(1,0,1/2)
φ_1	(0,0,0)	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,0)
φ_2	(0,0,0)	(0,0,1)	(0,0,1)	(0,1,0)	(0,1,1)	(0,1,1)	(1,0,0)	(1,0,1)

4. Conclusiones y trabajo futuro

Se ha diseñado un paquete de software que permite trabajar con posets, reticulados y álgebras de Łukasiewicz. Realiza cómputos automatizados que representan un ahorro de tiempo significativo y presenta los resultados en forma amigable. Al funcionar como una extensión del lenguaje Matlab, el usuario puede crear programas complejos que utilicen este paquete como herramienta.

Este software ha sido utilizado extensivamente por nuestro grupo de trabajo en álgebra de la lógica para la verificación de hipótesis. Esta experiencia ha servido para probar los algoritmos e incorporar mejoras. Los usuarios de nuestro grupo han proporcionado además valiosas sugerencias.

Se planea incluir nuevas funcionalidades para construcciones frecuentes en la teoría de reticulados, por ejemplo, la obtención del poset dual P de un reticulado L, esto es, el poset P tal que $D(P) = L$. También sería de utilidad definir nuevas clases en la estructura actual, que correspondan a otras categorías como la de álgebras de Heyting o de Boole. Un gran beneficio para el usuario sería poder incorporar fácilmente en sus documentos los diagramas de Hasse generados. A tal fin, implementaremos métodos para generar archivos METAPOST a partir de los diagramas de Hasse.

Se planea mejorar la interfaz de usuario especialmente en el aspecto gráfico. Por ejemplo, el usuario podría seleccionar elementos de un poset directamente sobre el diagrama de Hasse.

5. Agradecimientos

Agradecemos especialmente a Laura Rueda y a Ana María Suardíaz por sus numerosas sugerencias.

Referencias

[1] R. Balbes and P. Dwinger, *Distributive Lattices*, University of Missouri Press, Columbia, MO, (1974).

[2] V. Boicescu, A. Filipoiu, G. Georgescu and S. Rudeanu, *Lukasiewicz-Moisil Algebras*, North Holland, (1991).

[3] R. Cignoli, *Moisil Algebras*, Notas de Lógica Matemática No 27, Instituto de Matemática, Universidad Nacional del Sur, Bahía Blanca,(1970).

[4] Gr. C. Moisil, *Notes sur les logiques non-chrysippiennes*, Ann. Sci. Univ. Jassy, 27(1941), 86-98.

[5] A. Monteiro, *L'aritmétique des filtres et les espaces topologiques*, Notas de Lógica Matemática No 29–30, Instituto de Matemática, Universidad Nacional del Sur, Bahía Blanca, (1974).

[6] L. Monteiro, *Algebras de Lukasiewicz trivalentes monádicas*, Notas de Lógica Matemática No 32, Instituto de Matemática, Universidad Nacional del Sur, Bahía Blanca, (1974).

[7] H. A. Priestley, *Ordered topological spaces and the representation of distributive lattices*, Proc. London Math. Soc., 3,24; (1972), 507–530.

[8] H. A. Priestley, *Ordered sets and duality for distributive lattices*, Ann. Discrete Math., 23, (1984), 39–60.

[9] C. O. Sicoe, *Sur les ideaux des algèbres Lukasiewicziennes polivalentes*, Rev. Roum. Math. Pures et Appl., 12(1967), 391–401.

[10] C. O. Sicoe, *On many-valued Lukasiewicz algebra*, Proc. Japan Acad., 43(1967), 725–728.