

Uso de tipos abstractos de datos TAD en la demostración de teoremas propios de la teoría de conjuntos

Pedro E. Vera, Claudia P. González
Universidad Pontificia Bolivariana Seccional Bucaramanga

Aurora I. Gáfar
Universidad Industrial de Santander

María A. Rodríguez.
Corporación universitaria de ciencia y desarrollo – UNICIENCIA

Fecha de Recepción: 26/06/15 – Fecha de Aceptación: 07/08/15

Resumen

La implementación que se presenta en este trabajo es una aplicación para la manipulación formal de tipo abstracto de datos (TAD); ofrece una serie de herramientas para el modelaje con TAD ecuacionales, con las cuales se puede especificar y verificar propiedades de la estructura conjunto que luego son implementadas de forma automática para la demostración de algunos teoremas propios de lógica de predicados.

Palabras clave: Tipo abstracto de datos TAD, teoremas, lógica de predicados, dialogo.

Abstract

The implementation presented in this paper is an application for formal manipulation of abstract data type (ADT); It offers a number of tools for modeling with TAD ecuacionales, with which you can specify and verify properties of the whole structure which are then implemented automatically to show some own theorems of predicate logic.

Keywords: TAD abstract data type, theorems, predicate logic, dialogue.

para su aprendizaje por parte de los estudiantes de distintos niveles educativos incluidos el universitario y de ser limitada desde un punto de vista epistemológico.

La demostración matemática tiene hoy en día un significado más abierto, menos formalista y junto al pensamiento estrictamente deductivo; se resalta también la necesidad de potenciar otros modos validativos de tipo empírico-deductivo, la formulación de conjeturas, los ejemplos y contraejemplos, los procesos de generalización, la falsedad, los procesos de prueba y error entre otros. Usualmente los profesores de matemáticas ofrecen dos razones como explicación a estas dificultades, una es la falta de conciencia de los estudiantes, la necesidad de desarrollar demostraciones en el aula de clase y la otra, consiste en la poca madurez lógica [6].

Al interés en los últimos años de la educación matemática por el tema de la enseñanza y aprendizaje de la demostración se suma la necesidad de establecer diversos significados de la demostración en diferentes contextos institucionales, situaciones como: la validación, explicación, argumentación y la prueba; se consideran pruebas discutidas (o argumentos) que sean aceptados en el corazón de la comunidad, o por una persona; es decir, situaciones que sirven para justificar la veracidad de un enunciado o la eficacia de una acción [15].

I. INTRODUCCIÓN

Para algunos matemáticos la concepción del modelo de demostración está ligada a una disciplina abstracta, cuyos teoremas se deducen de conjuntos establecidos de axiomas, mediante razonamientos estrictamente lógicos y, en la práctica, la interpretación dominante es la consideración de la demostración como una argumentación deductiva y formal [1 -5]. En enunciados previos se ha cuestionado la aplicación de la demostración formal, dadas las dificultades

II. METODOLOGÍA. [1-3]

La exploración de la teoría matemática puede ser explicada en términos de la noción de “situación de exploración documental”, nuestra visión de situaciones de exploración es caracterizada por los siguientes parámetros: una colección C de conceptos conocidos, una colección K de hechos que describen completamente la interacción a la cual pertenecen los conceptos conocidos, un nuevo concepto N,

axiomas A que relacionan un nuevo concepto con uno conocido y una colección G de proposiciones objeto que exploran completamente la interacción entre el nuevo concepto y los conceptos conocidos, la notación de conocimiento completo no significa necesariamente una definición formal.

La investigación se fundamenta en documentos, específicamente artículos en donde se enuncia que la exploración completa es un esquema de abajo hacia arriba; el instrumento importante es la observación con las siguientes características: Un probador fijo que determina para cada base de conocimiento un conjunto fijo de fórmulas comprobables. Sin embargo, sin cambiar el probador, la colección de fórmulas comprobables puede aumentar drásticamente si nosotros agregamos fórmulas demostradas a la base de conocimiento. Similarmente, la longitud de pruebas puede disminuir drásticamente por la misma medida. También, el orden en que se demuestran las fórmulas (y si lo demostrado, se agrega a la base de conocimiento) puede drásticamente influir en el poder demostrador del probador. Se puede agregar un mecanismo al sistema de prueba que automáticamente agregue fórmulas probadas al conocimiento, si las secuencias de proposiciones objeto son las entradas a un probador del sistema en lugar de las proposiciones objeto aisladas. En *THEOREM* $\forall$, este mecanismo se lleva a cabo como una opción (el 'ExpandKnowledge') que puede ser un conjunto en el llamado de la prueba. El propósito de este diseño, es el aporte de nuevos conocimientos encaminados a la solución de un problema práctico.

Revisión Literaria: hace referencia al nivel más alto de desarrollo conseguido en un momento determinado en cada actividad como es la de combinar la funcionalidad de sistemas de software matemáticos y el sistema de pruebas en cualquier software, técnica o campo científico.

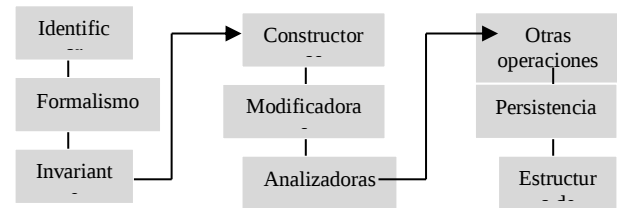
Abstracciones: La abstracción consiste en el aislamiento conceptual de una propiedad de un objeto (el probador PCS, Prueba-Compute-Solucion), también es el principio por el cual se aísla toda aquella información que no resulta relevante a un determinado nivel de conocimiento.

Sistemas y diseño de pruebas: Para diferentes aproximaciones en la axiomatización, hay formas diversas de cómo integrar el probador de teoría de conjuntos en el sistema propuesto:

Se restringe el lenguaje para el cual el probador es aplicable en lugar de tratar de soportar la disponibilidad del lenguaje enteramente para la teoría de conjuntos en el sistema propuesto.

Se adapta la semántica de la membresía y se desvía no significativamente desde la semántica intuitiva de los constructores de los lenguajes bien conocidos.

Figura 1. Pasos para el diseño de una estructura algebraica tipo abstracto [3]



Por último se introducen conceptos de ligamento dentro del sistema, obedeciendo a los tipos en todos los conjuntos constructores de los lenguajes teóricos ya sea inmediatamente o a nivel de sintaxis o a nivel de regla de inferencia. Esto puede sin embargo requerir un diseño fundamental del sistema completo y decidir no continuar este camino por el momento.

III. ESPECIFICACIÓN DE LA ESTRUCTURA ALGEBRAICA EN EL PROBADOR DE TEOREMAS

La estructura algebraica se compone de un conjunto de elementos, llamados el tipo de interés (tdi) y una funciones u operaciones que actúan sobre el tdi [1]. La definición del tdi y las funciones, en el probador de teoremas, se lleva a cabo de forma funcional recurrente o inductiva y las funciones se dotan de semántica mediante un conjunto de axiomas.

En el probador de teoremas, una estructura algebraica se define por medio de sus signature y sus axiomas. La signature es la especificación de la sintaxis del tipo, en la que se da nombre y aridad a las funciones. Los axiomas son un conjunto de igualdades con las cuales se dan la semántica a las operaciones de la estructura algebraica. En el siguiente cuadro se hace un resumen del concepto de estructura algebraica ecuacional [10]:

Tabla 1. Resumen del concepto de estructura algebraica ecuacional.

- Una Estructura algebraica (EA) es una estructura heterogénea.
 - $EA = \langle \text{Signature}, \text{Axiomas} \rangle$
 - La **signature** es una declaración de operaciones o funciones y sus funcionalidades
 - Los **axiomas** son predicados que condicionan las operaciones del EA.
- En un EA **ecuacional**, todos los axiomas son ecuaciones

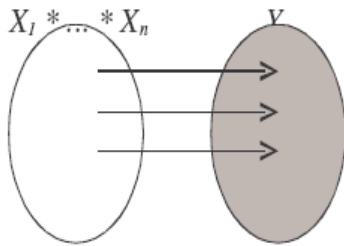
Para la estructura algebraica *EA* $Y(\text{con parametros } X_1, X_2, \dots, X_n)$ Podemos clasificar las funciones que lo especifican, de la

siguiente manera:

Iniciales: Definen los elementos básicos de la EA. Una función inicial puede tener aridad cero, en cuyo caso se trata de una constante, o tomar como parámetro variable de los tipos X_i . Su patrón instruccional es:

$$(signatura\ i : X_{i1} * X_{i2} * \dots * X_{im} \not\subset Y, 1 \leq i_j \leq n)$$

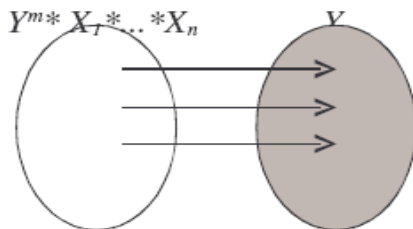
Figura 2. Función inicial



Constructoras: Utilizadas para definir elementos de Y , con base a las operaciones ya definidas y los elementos de tipo X_i . El patrón instruccional es:

$$(signatura\ c : Y * Y * \dots * Y * X_{i1} * X_{i2} * \dots * X_{im} \not\subset Y, 1 \leq i_j \leq n)$$

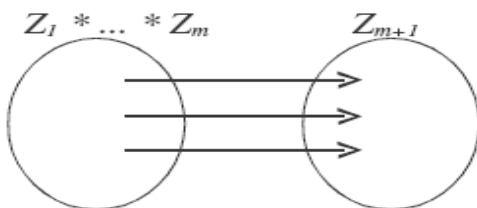
Figura 3. Función constructora



Selectoras: Funciones para manipular elementos de Y , junto con los de otras EA ya definidos. Las selectoras son la funcionalidad del tipo y permiten relacionarlo con las demás EA. El patrón instruccional es:

$$(signatura\ s : Z_1 * Z_2 * \dots * Z_m \not\subset Z_{m+1})$$

Figura 4. Función selectora



Los tipos Z_i se utilizan para indicar que puede ser de cualquier tipo del ambiente, que no guarda relación alguna

con la EA y tampoco con los parámetros X_i

Para las selectoras debe darse un conjunto de axiomas que sirven como reglas de reescritura para los valores de las EA, los axiomas determinan la igualdad entre dos expresiones, donde a la izquierda debe ser de la forma $f(x_1, x_2, \dots, x_m)$ siendo f una selectora de la EA y x_i una variable o valor del tipo especificado en la signatura para el i -ésimo parámetro de f y la expresión de la derecha debe ser una expresión del tipo f .

Figura 5. Notación para EAs

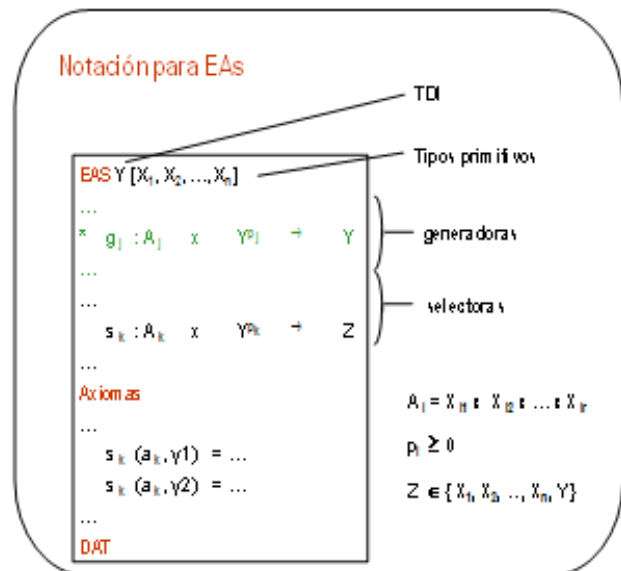
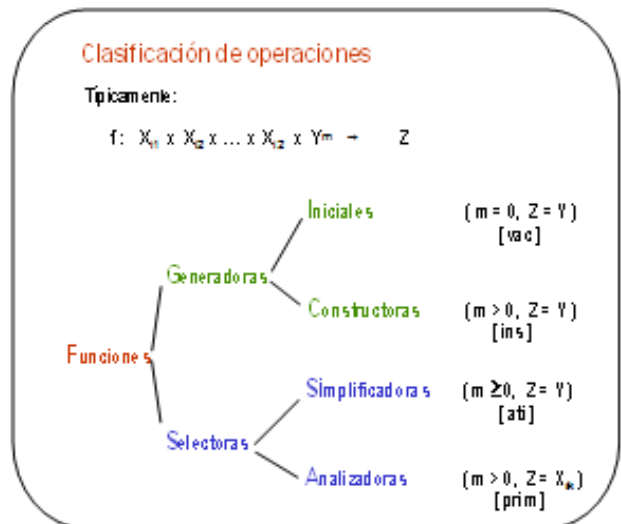


Figura 6. Clasificación de operaciones sobre AES



En la figura 5 se ilustra un resumen de los patrones instruccionales especificados anteriormente mediante

Notación para EAs presentados en la figura 5 y Clasificación de operaciones sobre AES mostrados en la figura 6 [1].

En la tabla 2 se presentan los indicadores numéricos para los nombres de los conjuntos que se emplean en la herramienta computacional y en la tabla 3 el menú del conjunto de operaciones básicas en la teoría de conjuntos y finalmente dos verificaciones formales formuladas en la teoría clásica de conjuntos.

Tabla 2. Tabla de indicadores numéricos para los nombres de los conjuntos

A =	1
B =	2
C =	3
D =	4
E =	5
F =	6

Tabla3. Menú del Conjunto de operaciones

1. Insertar elementos a un conjunto.
2. Eliminar elementos a un conjunto.
3. Indagar si x pertenece a un conjunto.
4. Dar la cardinalidad de un conjunto.
5. Contendencia.
6. Intersección entre dos conjuntos.
7. Unión entre dos conjuntos.
8. Igualdad entre dos conjuntos.
9. Diferencia parcial entre dos conjuntos.
10. Información de conjuntos.

Primer ejemplo de teorema a demostrar.

1. Verifique que: $A \Delta B = (A \cup B) - (A \cap B)$.

Para el programador:

Solución en el sistema (interactivo):

Siga los siguientes pasos:

1. Digite 1, luego digite el indicador para el conjunto A que aparece en pantalla.
2. Digite 1, luego digite el indicador para el conjunto B que aparece en pantalla.
3. Digite 7 para realizar la unión entre A y B, luego digite el indicador para el conjunto C para que usted vea que los elementos pertenecientes a el conjunto C es el resultado entre la unión entre A y B.
4. Digite 6 para realizar la intersección entre los conjuntos A y B, luego digite el indicador para el conjunto D para que usted vea que los elementos pertenecientes al conjunto D es el resultado de la intersección entre A y B.

5. Digite 9 para realizar la diferencia entre dos conjuntos, luego digite los indicadores para C y D que aparecen en pantalla, después asigne el indicador de E donde quedará guardado el resultado.

6. Digite 10 donde usted podrá ver la verificación propuesta.

Segundo ejemplo de teorema a demostrar.

2. Verifique que:

Si $A \subseteq B$ y B y C son disyuntos, entonces $A \cap (B \cup C) = A$.

Para el programador:

Conjunto A,B,C,D;

A=crearconj(1,50);

B=crearconj(1,50);

C=crearconj(1,50);

D=crearconj(1,50);

if (contendencia(A,B) && esvacio(D=interconj(B,C))

igual(A,inter(A,D));

info(inter(A,D);

info(A);

Solución en el sistema (interactivo):

Siga los siguientes pasos:

1. Digite 1, luego digite el indicador para el conjunto A que aparece en pantalla.
2. Digite 1, luego digite el indicador para el conjunto B que aparece en pantalla.
3. Digite 1, luego digite el indicador para el conjunto C que aparece en pantalla.
4. Digite 5, luego digite los indicadores para A y B en la pantalla aparecerá el indicador del resultado de la contendencia 1 para valor de verdad y 0 para valor de falsedad.
5. En el caso que el resultado sea 1 continúe con el paso 6, si el resultado es 0 continúe con el paso 12.
6. Digite 6, luego digite los indicadores para B y C, luego digite el indicador para D donde se encontrarán los elementos que pertenecen a esta intersección; en la pantalla usted podrá ver el resultado del conjunto intersección.
7. Digite 10, luego digite el indicador del conjunto D que le establecerá una función booleana que tiene como significado lo siguiente: Si adentro de las llaves no aparecen elementos el conjunto es vacío continúe con el paso 8, si usted ve entre las llaves elementos continúe con el paso 12.
8. Digite 7, luego digite los indicadores para B y C, luego digite el indicador para E donde se encontrarán los elementos pertenecientes a esta unión.
9. Digite 6, luego digite los indicadores para A y E, luego digite el indicador para F donde se

encontrarán los elementos pertenecientes a esta intersección.

10. Digite 8, luego digite los indicadores para A y F, en la pantalla aparecerá el indicador del resultado de la igualdad, 1 para el valor de verdad y 0 para el valor de falsedad.
11. Si en su pantalla aparece 1 se concluye que la verificación del enunciado tiene un valor de verdad. Dando por terminada la verificación, si no continúe con el siguiente paso.
12. Si en su pantalla apareció para la contención un valor de cero y en la intersección entre B y C un conjunto no vacío es decir con elementos, significa que el antecedente tiene un valor falso que implica que el consecuente tenga un valor falso.

Nota: Con la falsedad de cualquiera de las proposiciones del antecedente mencionadas anteriormente implica que el consecuente sea falso.

IV. CONCLUSIONES

Es significativo el papel del desarrollo matemático que se hace en la prueba de teoremas, se trata del aprovechamiento de la exactitud con que fue planeada la realidad, para deducir propiedades de esta mediante un método matemático con el beneficio de poder ayudarlo con el computador.

La motivación para indagar sobre la verificación de teoremas se basa en la necesidad de comparar el comportamiento del modelo, resultado de la abstracción y la realidad modelada en los aspectos que el diseñador considera relevantes.

Cuando se define estructuras algebraicas, se garantiza que los axiomas constituyen un sistema de reescritura terminante, lo cual es cierto para los lemas, por esta razón es necesario disponer de una estrategia que asegure que la aplicación de lemas no conduce a un ciclo interminable de reescritura de expresiones.

Diseñar el manejo de la persistencia. Básicamente se deben agregar dos operaciones: una constructora que tome la información de la memoria secundaria y cree un objeto del TAD, y otra que lleve el estado del objeto al disco en algún formato coherente con la primera operación.

Enriquecer el TAD con operaciones interesantes para los estudiantes. Casi siempre es conveniente ofrecer al estudiante, además de las operaciones indispensables, un buen conjunto de operaciones de frecuente uso. Aunque estas operaciones se pueden escribir en términos de operaciones más sencillas del TAD, por eficiencia, satisfacción del estudiante y extensibilidad del software, es conveniente agregarlas.

REFERENCIAS

- [1].Aho,A.,Hopcroft, J., Ullman, J., “Data Structures and Algorithms”,Cap.4, Addison-Wesley,1983.
- [2].Bauer, A., Clarke, E.M., and Zhao, X. *Analytica: An Experiment in Combining Theorem Proving and Symbolic Computation*. AISMC-3, Steyr, Austria : In international Conference Artificial Intelligence and Symbolic Mathematical Computation., 1996. pp. pp 21-37.
- [3].Bergin, J., “Data Abstraction: The object-Oriented Approach Using C++”, Cap.12, McGraw-Hill,1994.
- [4].Bertoli, P. G., Calmet, J., Giunchiglia, F.; and Homann, K. Specification and integration of theorem Provers and computer Algebra Systems. *Fundamenta Informaticae*. (1999).
- [5].Buchberger,B.,Jebelean,Triftner,F.,Marin,M.,Tomuta,E., and Vasaru,D. A Survey of the theorema Project. In *Proceedings of ISAAC'97. International Symposium on Symbolic and Algebraic Computation*. Maui,Hawaii.W.Kuechlin: s.n,1997.pp.384-391.
- [6].Dickson, Linda., Brown,M.,Olwen G.,El Aprendizaje de las Matematicas”, Barcelona Labor, 1991.
- [7].Feldman, M., “Data Structures with Modula-2”, Cap.8,Prentice-Hall,1988.
- [8].Gutierrez Raul ,Modulo Tutor Inteligente como aplicacion pedagogica de la ley distributiva,Universidad de Valle (colombia), Revista enseñanza y Tecnologia septiembre-diciembre 2000-7.
- [9].Horowitz,E., “Fundamentals of Data Structures”, Cap.9.Computer Science Press,1983.
- [10].Knuth, D., “The Art of Computer Programming”, Vol.3-Sorting and Searching,(pp. 506-549),Addison-Wesley,1973.
- [11].Kruse,R., “Data Structure & Program Design”, 2da.edición,Cap.6, Prentice-Hall,1986.
- [12].Lipschutz, S., “Estructura de datos”,Cap.9, Prentice-Hall,1986.
- [13].Tenenbaum,A.,Langsam,Y., “Estructuras de datos en C”, Cap.7, Prentice Hall, 1993.
- [14].Tremblay,J.,Sorenson,P., “An introduction to data Structures with Applications”,Cap.6,McGraw-Hill,1976.
- [15].Varon Gustavo. Elementos de la Matematica Intuitiva , Bedut , 1969

[16].Wirth,N., “Algorithms & Data Structures”, Cap.5, Prentice-Hall, 1986.

[17] González, José. Apuntes de Lógica Matemática . Razonamientos y Demostraciones. Universidad Cádiz. España. Departamento de Matemáticas. Abril 2015.

[18] Labra, José. Lógica de predicados. Escuela universitaria de Ingeniería técnica en Informática de Oviedo (E.U.I.T.I.O). 2015.

[19]Serrano, Montserrat. Estructura de datos. Departamento de Informática. Campus de Segovia. Universidad de Valladolid. 2014.

[20] Maia, Manuel. Lógica Proposicional, Teoremas y demostraciones. Marzo 2012.

modelación Matemática en áreas de la salud, educación, genero.



Maria Alejandra Rodríguez Duarte, Lugar de nacimiento Bucaramanga Colombia, Economista de la Universidad Industrial de Santander, Especialista en Gestión Pública de la Escuela Superior de Administración Pública-ESAP, Magíster en Derechos humanos de la Universidad Industrial de Santander, Asesora en investigación de la Corporación Universitaria de Ciencia y Desarrollo UNICIENCIA, extensión Bucaramanga. Lider del grupo de investigación en desarrollo y crecimiento GIDEC, areas de investigación en derechos humanos, género y sociedad, competitividad y mercado laboral, desarrollo regional, planificación y territorio.

BIOGRAFÍA



Pedro Elías Vera Bautista, Lugar de nacimiento Pamplona Colombia, Licenciado en Matemática e Informática Educativa, Especialista en Gestión de Proyectos informáticos de la Universidad de Pamplona, Magíster en Matemáticas de la Universidad Nacional Experimental del Táchira UNET Venezuela, Profesor Asociado del Departamento de Ciencias Básicas de la Universidad Pontificia Bolivariana Seccional Bucaramanga. Areas en investigación en modelación Matemática, Nuevos materiales, Dispositivos móviles inteligentes.



Claudia Paulina González Cuervo, Lugar de nacimiento Tunja, Licenciada en Ciencias de la Educación especialidad en Física y Matemáticas de la Universidad Pedagógica y Tecnológica de Colombia, Tunja, Magíster y Doctorado en Ciencias Física de la Universidad del Valle, Cali. Profesor Asociado del Departamento de Ciencias Básicas de la Universidad Pontificia Bolivariana Seccional Bucaramanga. Áreas en investigación en Nanopartículas, aleaciones, espectroscopia Mössbauer, difracción de rayos x, Microscopia electrónica, Nuevos Materiales.



Aurora Inés Gáfaró Rojas, Lugar de nacimiento Pamplona Colombia, Licenciada en Matemática y Computación, Especialista en Educación Matemática de la Universidad de Pamplona, Doctorado en Estadística, Matemática e Informática de la Universidad Pública de Navarra España. Profesora Asociada del departamento de salud pública escuela de medicina de la facultad de salud Universidad Industrial de Santander UIS Bucaramanga. Areas en investigación en