

Deducción automática en anillos ternarios: Algunos métodos de procesamiento del conocimiento matemático

J.A. Alonso-Jiménez* J. Borrego-Díaz † A. Chávez-González
Dpto. Ciencias de la Computación e Inteligencia Artificial. Universidad de Sevilla
C/Tarfia s/n 41012, Sevilla e-mail: jborrego@cica.es

Abstract

En este trabajo se presenta la demostración automatizada de diversos resultados relativos a los anillos ternarios planos *naturales* [Klucky 95]. Este ejemplo motiva la utilización de diversas herramientas de la Lógica Computacional para el procesamiento de los axiomas, y la obtención automática de demostraciones, así como la presentación de varios problemas de investigación básica relativos a la gestión del conocimiento matemático con técnicas de Inteligencia Artificial, basadas en la Lógica Matemática.

Area: Ciencias de la Computación e Inteligencia Artificial

Palabras clave: DEMOSTRACIÓN AUTOMÁTICA, OTTER, GEOMETRÍAS AXIOMÁTICAS

1 Introducción

La demostración automática de teoremas es el área de las Ciencias de la Computación e Inteligencia Artificial que pretende alcanzar la meta, propuesta por Leibniz, de un cálculo universal para la obtención de pruebas mecánicas en Matemáticas. La utilidad de los sistemas obtenidos no se limita a las Matemáticas; se aplican a problemas de Inteligencia Artificial, diseño de circuitos, verificación automática de programas, y, en general, a cualquier tipo de problemas que requiera razonamiento. El término *razonamiento automático*, introducido en 1980, engloba tanto el diseño de estos programas como otros programas relacionados (p.e. los generadores automáticos de modelos), y sus aplicaciones [Wos 85].

Los primeros programas diseñados para la obtención de pruebas mecánicas datan de los años cincuenta. Pero no es hasta los años sesenta, con la aplicación directa de técnicas clásicas de la Lógica Matemática (que se remontan a trabajos de Gödel, Skolem y Herbrand), cuando se obtiene un salto cualitativo en el desarrollo de estos sistemas (entre otros, [Davis y Putnam 60], [Robinson 65]). A partir de estos trabajos, se han desarrollado múltiples técnicas y refinamientos que han mejorado sustancialmente los demostradores automáticos, hasta el punto de demostrar cuestiones abiertas en Matemáticas¹.

El carácter Lógico–Matemático de tales sistemas (basados en lógicas de primer orden o superior), junto con la gestión eficiente de los distintos motores de inferencia, hace que

*Financiado parcialmente por el proyecto DGES PB96-0098-C04-04 y PAI TIC-137.

†Financiado parcialmente por el proyecto DGES PB96-1345 y PAI TIC-137.

¹Quizás el más importante sea la demostración de la conjetura de Robbins [McCune 97] sobre álgebras de Boole, una cuestión profundamente estudiada por Tarski y propuesta hace unos sesenta años.

su uso, como asistente de investigación en Matemáticas, sea complejo: es necesario conocer los mecanismos de deducción de nuevos resultados, como se generan y/o descartan, para orientar, de algún modo, la búsqueda de la prueba. Esta circunstancia es la diferencia fundamental entre estos sistemas y cualquier otro lenguaje de programación; el usuario no gestiona directamente la ejecución del programa, y es, por tanto, un *nuevo nivel* de programación.

En el trabajo introducimos diversas técnicas utilizadas en la demostración automática de teoremas, usando como guía de la exposición las pruebas que hemos obtenido, con un demostrador automático, de algunos de los resultados de [Klucky 95], justificando la corrección de algunas de las técnicas empleadas.

2 Anillos ternarios planos naturales

Los anillos ternarios, introducidos por Marshall Hall [Hall 43], representan una fundamentación algebraica para diversas geometrías axiomáticas. Partiendo del estudio de las propiedades de tales geometrías, abstraer su representación utilizando una operación ternaria $t: R^3 \rightarrow R$ (donde R es un conjunto en correspondencia biyectiva con todas las rectas que pasan por un punto, salvo una). Por ejemplo, en el plano real, la ecuación $y = t(x, m, b)$ representa la recta de $y = mx + b$. Existe una correspondencia entre los planos afines y las siguientes estructuras definidas en el lenguaje $L = \{t, \mathbf{0}, \mathbf{1}\}$ ($\mathbf{0}, \mathbf{1}$ constantes):

Definición 1 *Un anillo ternario plano es una estructura $(R, \{t, 0, 1\})$, con $0 \neq 1$, $t: R^3 \rightarrow R$ verificando las siguientes propiedades:*

1. $\forall a \forall b \forall c [t(0, b, c) = c \wedge t(a, 0, c) = c]$.
2. $\forall a [t(a, 1, 0) = a \wedge t(1, a, 0) = a]$.
3. Si $b, b', m, m' \in R$ con $m \neq m'$, entonces el siguiente sistema tiene solución única:

$$\begin{cases} t(x, m, b) = y \\ t(x, m', b') = y \end{cases}$$

4. Si $a, a', b, b' \in R$, con $a \neq a'$, entonces el siguiente sistema tiene solución única:

$$\begin{cases} t(a, x, y) = b \\ t(a', x, y) = b' \end{cases}$$

5. Para cualesquiera a, m, c , la ecuación $t(a, m, x) = c$ tiene una única solución.

Se prueba que todo plano afín define un anillo ternario plano (previa elección de coordenadas), y recíprocamente, todo anillo ternario define un plano afín (una prueba se puede encontrar, por ejemplo, en [Blumenthal 61])². Por tanto, este tipo de estructuras interpreta, de manera algebraica, este tipo de geometrías. El objeto de investigación en este trabajo será una variante de los anillos ternarios planos, introducida en [Klucky 95]:

Definición 2 *Una estructura $\langle M, t, *, \{0_L, 0_R\} \rangle$ es un anillo ternario plano natural (NPTR) si verifica las siguientes condiciones:*

² Axiomáticamente, un plano afín es una estructura de incidencia donde cualesquiera dos puntos distintos están conectados por una única recta; dado un punto P y una recta r no incidente con él, existe una única recta paralela (disjunta) a r e incidente con P ; y existen cuatro puntos no colineales tres a tres.

- (A) $\forall x, y, m \exists! b [t(x, m, b) = y]$.
 (B) $\forall m, m', b, b' [m \neq m' \rightarrow \exists! x (t(x, m, b) = t(x, m', b'))]$.
 (C) $\forall x, x', y, y' [x \neq x' \rightarrow \exists! m, b (t(x, m, b) = y \wedge t(x', m, b) = y')]$.
 (D) $\forall m, b [t(0_L, m, b^*) = b] \wedge \forall x, b [t(x, 0_R, b^*) = b]$.
 (E) *La función $(\cdot)^*$ es biyectiva.*

3 Preprocesamiento

Para utilizar un demostrador automático de manera eficiente, es necesario reescribir los axiomas que definen a los NPTR³. De hecho, salvo el axioma (E), todos los demás axiomas están escritos como fórmulas en el lenguaje de primer orden $L_0 = \{t, *, 0_L, 0_R\}$, con $t, (\cdot)^*$ símbolos de función de aridades 3 y 1, respectivamente, y 0_L y 0_R símbolos de constante. Es trivial escribir una fórmula para (E), pero antes de hacerlo, es preferible describir qué tipo de transformación aplicaremos a estos axiomas.

Una primera etapa consiste en obtener una nueva axiomatización de la teoría, simplificando la complejidad de los axiomas, tanto de los cuantificadores como de la estructura interna de la fórmula. Con respecto a los cuantificadores se utilizan las *funciones de Skolem*. Intuitivamente, una función de Skolem representa la dependencia funcional de una variable con respecto a otras, y su uso se fundamenta en el siguiente resultado clásico:

Teorema 3 *Sean T una teoría de primer orden, $\varphi(\vec{x}, y)$ una fórmula y f un nuevo símbolo de función n -ario. Si T' se obtiene añadiendo a T el axioma*

$$\forall \vec{x} [\exists y \varphi(\vec{x}, y) \rightarrow \varphi(\vec{x}, f(\vec{x}))]$$

entonces la teoría T' es una extensión conservativa de T , es decir, T y T' demuestran los mismos teoremas en el lenguaje de T .

Aplicando adecuadamente el resultado a los axiomas de una teoría, se obtiene una extensión conservativa cuyos axiomas son fórmulas sin cuantificadores existenciales. Una vez eliminados éstos, escribimos la matriz de cada fórmula en forma normal conjuntiva (conjunción de disyunciones de literales). En nuestro caso (teniendo en cuenta que $\exists! x \varphi$ abrevia la fórmula $\exists x [\varphi(x) \wedge \forall y (\varphi(y) \rightarrow x = y)]$), la extensión conservativa que obtenemos es la teoría $NPTR_0$, formada por las siguientes disyunciones (llamadas cláusulas)⁴:

- $A_E : t(X, M, ta(X, M, B)) = B$
 $A_U : (t(X, M, Y) \neq B) | (ta(X, M, B) = Y)$
 $B_E : (M1 = M2) | (t(tb(M1, B1, M2, B2), M1, B1) = t(tb(M1, B1, M2, B2), M2, B2))$
 $B_U : (M1 = M2) | (t(X, M1, B1) \neq t(X, M2, B2)) | (tb(M1, B1, M2, B2) = X)$
 $C_{E,1} : (X1 = X2) | (t(X1, tcm(X1, X2, Y1, Y2), tcb(X1, X2, Y1, Y2)) = Y1)$
 $C_{E,2} : (X1 = X2) | (t(X2, tcm(X1, X2, Y1, Y2), tcb(X1, X2, Y1, Y2)) = Y2)$
 $C_{U,1} : (X1 = X2) | (t(X1, M, B) \neq Y1) | (t(X2, M, B) \neq Y2) | (tcm(X1, X2, Y1, Y2) = M)$
 $C_{U,2} : (X1 = X2) | (t(X1, M, B) \neq Y1) | (t(X2, M, B) \neq Y2) | (tcb(X1, X2, Y1, Y2) = B)$
 $D_1 : t(0_L, M, asterisco(B)) = B$
 $D_2 : t(X, 0_R, asterisco(B)) = B$
 $E_1 : inversa(asterisco(B)) = B$
 $E_2 : asterisco(inversa(B)) = B$

Las dos últimas fórmulas afirman la existencia de la inversa (representada por *inversa(.)*) de la función $(\cdot)^*$ que la representada por comodidad como *asterisco(.)*

³Esto no es estrictamente necesario, pues es automatizable, y de hecho, OTTER lo hace

⁴Las cláusulas de tipo *E* son de *existencia de solución*, las de tipo *U* son de *unicidad*, el símbolo $|$ denota la disyunción y \neq es el símbolo de negación.

4 Mecanismos de inferencia

El siguiente paso para trabajar con la teoría es analizar la estructura de sus axiomas, para elegir las reglas de inferencia adecuadas. Como es una teoría con igualdad, junto a las reglas de inferencia propias de la lógica de predicados, necesitamos reglas para el razonamiento con igualdad. Entre otras, algunas de las reglas que utilizamos son:

- RESOLUCIÓN BINARIA:

$$\frac{C_1 | \dots | C_k | \dots | C_n, \quad D_1 | \dots | D_j | \dots | D_m}{(C_1 | \dots | C_{k-1} | C_{k+1} | \dots | C_n | D_1 | \dots | D_{j-1} | D_{j+1} | \dots | D_m) \sigma}$$

donde σ es un unificador de máxima generalidad de C_k y el complementario de D_j .

- HIPERRESOLUCIÓN:

$$\frac{M_1, \dots, M_k, \quad -A_1 | \dots | -A_k | B_1 | \dots | B_m}{(B_1 | \dots | B_m) \sigma}$$

(siendo M_i, A_i, B_i átomos), donde σ es un unificador de máxima generalidad para M_j y A_j ($1 \leq j \leq k$).

- RESOLUCIÓN UR: Es similar a la hiperresolución, pero devuelve un único literal.
- DEMODULACIÓN: Consiste en utilizar una igualdad como operador de reescritura: Si tenemos la igualdad $t_1 = t_2$ orientada, sustituimos en la fórmula toda instancia de t_1 por la respectiva instancia de t_2 .
- PARAMODULACIÓN:

$$\frac{L_1 | \dots | L_i(t_1) | \dots | L_n, \quad M_1 | \dots | M_{j-1} | (t_2 = t_3) | M_{j+1} | \dots | M_k}{(L_1 | \dots | L_i(t_3) | \dots | L_n | M_1 | \dots | M_{j-1} | (t_2 = t_3) | M_{j+1} | M_k) \sigma}$$

donde σ es un unificador de máxima generalidad de t_1 y t_2 , y $L_i(t_3)$ se obtiene sustituyendo toda estancia de t_1 en L_i por t_3 (módulo unificador).

Estas reglas forman un conjunto *refutacionalmente completo* para la lógica clausal de primer orden con igualdad.

5 El demostrador automático OTTER

El demostrador que utilizamos en este trabajo es OTTER [McCune 94], desarrollado en el Argonne National Laboratory, uno de los más potentes, y que admite un amplio conjunto de reglas de inferencia, estrategias de búsqueda, etc.

OTTER demuestra por refutación, es decir, dado un conjunto de fórmulas, intenta deducir una inconsistencia. El esquema de su funcionamiento se basa en la estrategia del conjunto soporte [Wos et al. 65]: Divide el conjunto de entrada en dos subconjuntos, las *usables* y las de *soporte*. En general, el conjunto de usables suele ser un conjunto consistente de fórmulas (por ejemplo, la teoría a estudiar) y el conjunto soporte el resto (usualmente, la negación del teorema a demostrar). La estrategia consiste en aplicar las reglas de inferencia eligiendo al menos una fórmula del conjunto soporte y añadiendo las conclusiones a ese conjunto (las reglas de inferencia son adecuadas, y por tanto no es posible deducir una inconsistencia a partir de un conjunto consistente) hasta vaciar el conjunto soporte (demostración no encontrada) o llegar a una inconsistencia. El usuario debe elegir las reglas de inferencia que se utilizarán y la partición del conjunto original. Hay que tener en cuenta que la estrategia del conjunto soporte no es completa (cualquier partición del conjunto de entrada no conduce a una inconsistencia). De las cláusulas así generadas, OTTER utiliza diversos mecanismos para retener las que son interesantes.

6 Teoremas demostrados

En esta sección comentamos las pruebas obtenidas con el demostrador. Salvo algunas excepciones importantes, una primera prueba del teorema la hemos obtenido mediante el modo autónomo `auto2` de OTTER, que es una opción que, al menos parcialmente, automatiza la elección de las reglas de inferencia y de la estrategia a seguir para encontrar la prueba. Una vez conseguida una prueba, hemos depurado la estrategia de búsqueda de ésta para encontrar una más simple. El trabajo ha sido realizado con la última versión UNIX/LINUX del programa, en un PC (AMD K6-400). Por cuestiones de espacio, sólo presentamos algunas características de dichas pruebas.

6.1 Unicidad de 0_L , 0_R

Las pruebas obtenidas de la unicidad de los elementos 0_L y 0_R en un NPTR son:

Teorema 4 *Los elementos 0_L y 0_R son únicos con esa propiedad.*

Demostración: Unicidad de 0_L : Encontrada por OTTER en 0.07 segundos:

```

2 [] M1=M2 | t(X,M1,B1) != t(X,M2,B2) | tb(M1,B1,M2,B2)=X.
3 [] t(o1,M,asterisco(B))=B.
4 [] o1!=o.
5 [] t(o,M,asterisco(B))=B.
8 [ur,4,2,3] tb(o1,asterisco(t(o1,o,A)),o,A)=o1.
12 [ur,5,2,4] tb(o1,asterisco(t(o,o,A)),o,A)=o.
16,15 [para-into,8.1.1.2.1,3.1.1] tb(o1,asterisco(A),o,asterisco(A))=o1.
19 [para-into,12.1.1.2.1,5.1.1,demod,16] o1=o.
21 [binary,19.1,4.1] $ F.
```

Las cláusulas no inferidas (las hipótesis, con []) son axiomas de $NPTR_0$ ó la negación del teorema. Por ejemplo, las cláusulas `o1!=o` y `t(o,M,asterisco(B))=B` afirman la existencia de otro cero a la izquierda. La prueba de la unicidad de 0_R obtenida es:

```

5 [] X1=X2 | t(X1,M,B) != Y1 | t(X2,M,B) != Y2 | tcm(X1,X2,Y1,Y2)=M.
7 [] t(X,or,asterisco(B))=B.
8 [] or!=o.
9 [] t(X,o,asterisco(B))=B.
19,18 [ur,8,5,7,7] tcm(or,o,A,A)=or.
94 [ur,9,5,8,9,demod,19] or=o.
96 [binary,94.1,8.1] $ F. ■
```

6.2 Propiedades del producto

A partir de la operación ternaria t se pueden definir una multiplicación y una suma en un NPTR. El producto se define como

$$a \cdot b = t(a, b, (0_L)^*)$$

se verifican las siguientes propiedades:

Proposición 5 *Sea M un NPTR. En las condiciones anteriores, se verifica que:*

1. Para cualesquiera $a, b \in M$, si $a \neq 0_L$, existe un único elemento x tal que $a \cdot x = b$.
2. Para cualesquiera $a, b \in M$, si $a \neq 0_R$, existe un único elemento x tal que $x \cdot a = b$.
3. Para cualesquiera $a, b \in M$ $a \cdot b = 0_L$ sii $a = 0_L$ ó $b = 0_R$.

Propiedad	Tiempo (s.)	Cl. generadas	Retenidas	Long. prueba
(1), unicidad	0.06	590	146	7
(1), existencia	0.02	145	58	9
(2), unicidad	0.01	55	20	6
(2), existencia	0.01	9	5	4
(3), \implies	0.02	56	27	6
(3), \impliedby	0.00	14	10	2

6.3 Propiedades de la suma

De la propiedad (1) de la proposición anterior se tiene que para cada $x \neq 0_L$ existe un único elemento $e(x) \in M$ tal que $x \cdot e(x) = x$. Si definimos $e(0_L) = 0_R$, la función $e : M \rightarrow M$ está bien definida (en $NPTR_0$) por las cláusulas

- $(e(0_L) = 0_R)$.
- $(X = 0_L) \mid (\text{prod}(X, e(X)) = X)$.
- $(X = 0_L) \mid (\text{prod}(X, Y) \neq X) \mid (e(X) = Y)$.

Así, definimos una nueva operación, $+$, como $a + b := t(a, e(a), b^*)$.

Proposición 6 Sea M un $NPTR_0$

1. Para todo $a \in M$, $a + 0_L = a$ y $0_L + a = a$.
2. Para cualesquiera $a, b \in M$ existe un único $x \in M$ tal que $a + x = b$.

Propiedad	Tiempo (s.)	Cl. generadas	Retenidas	Long. prueba
(1) $a + 0_L = a$	0.01	9	6	5
(1) $(0_L + a) = a$	0.00	3	3	1
(2), existencia	0.02	95	83	7
(2), unicidad	0.01	98	31	7

6.4 Búsqueda de soluciones

Podría parecer que, con las demostraciones por refutación, se pierde cierta información que es interesante. Por ejemplo, en la propiedad (2) de la proposición 5,

$$\forall a[a \neq 0_R \rightarrow \exists!x(x \cdot a = b)]$$

puede ser muy útil conocer la expresión de x con respecto a los elementos a y b . Existe una primera dificultad para resolver esta cuestión, y es si tal expresión existe: Demostrando la existencia de un elemento no nos asegura que éste se pueda expresar de manera simple en el lenguaje utilizado. En el caso que nos ocupa, esta propiedad sí es cierta; es consecuencia del *Teorema de Herbrand*:

Teorema 7 Sea T una teoría de primer orden cuyo axiomas son fórmulas abiertas, y sea $\varphi(x)$ una fórmula abierta. Son equivalentes:

- $T \vdash \exists x\varphi(x)$
- Existen t_1, \dots, t_n términos cerrados tales que $T \vdash \varphi(t_1) \vee \dots \vee \varphi(t_n)$.

Corolario 8 Existe un L_0 -término $\mathbf{t}(u, v)$ tal que $NPTR_0 \vdash u \neq 0_R \rightarrow \mathbf{t}(u, v) \cdot u = v$

Demostración: Para aplicar el teorema anterior a la teoría $NPTR_0$, expandimos el lenguaje de tal teoría con dos nuevas constantes, a, b y tomamos la teoría T en el nuevo lenguaje cuyos axiomas son los de $NPTR_0$. Por tanto,

$$T \vdash \exists x[a \neq 0_R \rightarrow (a \cdot x = b)]$$

Aplicando el teorema anterior, existen términos cerrados t_1, \dots, t_n tales que

$$T \vdash (a \neq 0_R \rightarrow t_1 \cdot a = b) \vee \dots \vee (a \neq 0_R \rightarrow t_n \cdot a = b)$$

pero como $T \vdash a \neq 0_R \rightarrow \exists! x[x \cdot a = b]$, utilizando los axiomas de igualdad, se concluye que

$$T \vdash a \neq 0_R \rightarrow t_1 \cdot a = b$$

El término t_1 es un término (cerrado) en el lenguaje $L_0 \cup \{a, b\}$. Si $\mathbf{t}(u, v)$ es el L_0 -término que se obtiene sustituyendo las constantes a y b por u y v respectivamente, entonces, por el teorema de constantes se verifica que $NPTR_0 \vdash u \neq 0_R \rightarrow \mathbf{t}(u, v) \cdot u = v$ ■

El corolario anterior nos asegura la existencia de una expresión de este tipo. La fundamentación de la obtención de respuestas es una generalización del concepto de respuesta utilizada en programación lógica (por ejemplo en PROLOG) [Kunen 96]. De hecho, OTTER tiene implementado una función, `$ans(X)`, que devuelve el valor obtenido para la variable X . La búsqueda del término $\mathbf{t}(u, v)$ sigue la idea de la demostración anterior: La entrada de OTTER son los axiomas de $NPTR_0$ junto con el siguiente conjunto soporte:

- `(a != or)`.
 - `(prod(X, a) != b) | $ans(X)`
- y (en 0.1 seg.) nos devuelve la solución⁵ `$ans(tb(or, asterisco(b), a, asterisco(ol)))`, es decir,

$$\mathbf{t}(u, v) = tb(0_R, v^*, u, (0_L)^*)$$

Interpretando este resultado en el lenguaje original de NPTR, el elemento x tal que $x \cdot a = b$, para $a \neq 0_R$, es la única solución de la ecuación $t(x, 0_R, b^*) = t(x, a, (0_L)^*)$.

6.5 Isotopías: Búsqueda de funciones

Veamos por último, cómo se puede estudiar la existencia de funciones usando el demostrador. Aparentemente, el razonamiento con funciones debería de utilizar una lógica de orden superior. Sin embargo, no es necesario este tipo de ampliaciones en general. Veamos un ejemplo. El teorema 4.2 de [Klucký 95] afirma que la propiedad de ser NPTR se conserva bajo isotopías. Una isotopía entre dos anillos ternarios planos (M, t_1) y (M, t_2) (para simplificar, con el mismo universo) es una 4-upla de permutaciones de M , $\alpha, \beta, \gamma, \delta$, tales que para cualesquiera $x, m, b \in M$

$$\delta(t_2(x, m, b)) = t_1(\alpha(x), \beta(m), \beta(b))$$

El teorema asegura que si $(M, t_1, *, 0_L, 0_R)$ es un NPTR, entonces $(M, t_2, \alpha^{-1}(0_L), \beta^{-1}(0_R))$ admite una estructura de NPTR, para una cierta aplicación $b \mapsto b^\times$

No vamos a describir la demostración, usando OTTER, de este hecho, sólo nos ocuparemos de la cuestión más importante, la existencia de la aplicación $b \mapsto b^\times$ tal que $(M, t_2, \alpha^{-1}(0_L), \beta^{-1}(0_R))$ es un NPTR. Para encontrar tal función, razonamos como en la subsección anterior, y tomamos como conjunto soporte las siguientes fórmulas:

$$\begin{aligned} \text{delta}(t_2(X, M, B)) &= \text{t}(\text{alfa}(X), \text{beta}(M), \text{gamma}(B)). \\ \text{inversa-delta}(\text{delta}(X)) &= X. \\ \text{delta}(\text{inversa-delta}(X)) &= X. \end{aligned}$$

⁵Esta no es la solución que ofrece OTTER en el modo auto2, que es más compleja. La que ofrecemos ha sido obtenida aplicando distintas estrategias para dirigir al demostrador hacia una simplificada. Como ya hemos justificado anteriormente, cualquier solución denota el mismo elemento, pues es único.

```

(alfa(inversa-alfa(X)) = X).
(inversa-alfa(alfa(X)) = X).
(beta(inversa-beta(X)) = X).
(inversa-beta(beta(X)) = X).
(gamma(inversa-gamma(X)) = X).
(inversa-gamma(gamma(X)) = X).
(alfa(o12) = o1).
(beta(or2) = or).
(all X ((all M (t2(o12 , M , X) = b)) -> $ans(X))).

```

La última de las cuales exige la respuesta para X que nos asegura que $\alpha^{-1}(0_L)$ es el cero a la izquierda. La respuesta obtenida es $\$ans(inversa-gamma(asterisco(delta(b))))$, es decir, $x^x = \gamma^{-1}(\delta(x)^*)$ es la aplicación pedida.

7 Conclusiones

Hemos mostrado el uso fundamentado de un demostrador automático para la investigación en Matemáticas, eligiendo como caso de estudio los anillos ternarios. El futuro trabajo reflejará, en el caso especial de este tipo de estructuras, los principales problemas a resolver en este campo, entre los que destacamos

- El diseño de estrategias para eliminar, durante el proceso, información redundante.
- Estrategias para conducir la prueba y obtener así demostraciones más simples.
- En el caso concreto de los anillos ternarios, buscar cotas para la complejidad de las cláusulas de una demostración en función del teorema-objetivo, de origen geométrico.

Referencias

- [Blumenthal 61] BLUMENTHAL, L.M.: *A modern view of Geometry*. Freeman, 1961.
- [Davis y Putnam 60] DAVIS, M.; PUTNAM, H.: *A computing procedure for quantification theory*. Journal of the ACM 7 (3), 201–215 (1960).
- [Hall 43] HALL, M.: *Projective Planes* Trans. of AMS 54, 229–277 (1943).
- [Klucký 95] KLUCKÝ, D.: *Isotopic Invariants of natural planar ternary rings*. Mathematica Bohemica 120 (3), 325–335 (1995).
- [Kunen 96] KUNEN, K.: *The Semantics of Answer Literals*. J. Automated Reasoning 17 (1996), 83–95.
- [McCune 94] MCCUNE, W.: *OTTER 3.0 Reference Manual and Guide*. Tech. Report ANL-94/6, Argone National Laboratory, 1994.
- [McCune 97] MCCUNE, W.: *Solution of the Robbins Problem*, J. Automated Reasoning 19 (3), 263–276 (1997).
- [Robinson 65] ROBINSON, J.A.: *A machine oriented logic based on the Resolution Principle*. J. of the ACM 12, 23–41 (1965).
- [Wos et al. 65] WOS, L.; ROBINSON, G.; CARSON, D.: *Efficiency and completeness of the set of support strategy in theorem proving*. J. of ACM 12 (4), 536–541 (1961).
- [Wos 85] WOS, L.: *What is Automated Reasoning?* J. of Automated Reasoning 1, 6–9 (1985).