ESTRATEGIAS PARA LA DEMOSTRACIÓN AUTOMÁTICA DE TEOREMAS

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

Resumen: *En este trabajo se presenta la utilización de un demostrador automático, OTTER, como asistente en la investigación de propiedades de ciertas estructuras matemáticas, los anillos ternarios planos* naturales*, introducidos en [Klucký 95]. Con este motivo se muestran diversas estrategias para conducir el procesamiento de la información contenida en los axiomas y la obtención de pruebas, mejorando incluso las demostraciones obtenidas por OTTER en su modo automático.*

1 Introducción

La demostración automática de teoremas es un campo de investigación en 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 en Matemáticas de forma mecánica. La utilidad de este tipo de programas no se limita a este tipo de demostraciones: las técnicas y sistemas obtenidos 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.

Sin embargo, 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 su uso, como asistente para la investigación en Matemáticas, sea complejo: es necesario conocer los mecanismos de deducción de nuevos resultados, cómo se generan y/o descartan, para orientar, de algún modo, la búsqueda automatizada 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.

El trabajo pretende introducir 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 [Klucký 95], justificando la corrección de algunas de las técnicas empleadas.

2 Anillos ternarios planos naturales

Los anillos ternarios fueron introducidos por Marshall Hall [Hall 43] como base para una fundamentación algebraica de las geometrías axiomáticas. Partiendo del estudio de las propiedades de tales geometrías, abstrae su representación utilizando una operación

*Financiado parcialmente por el proyecto DGES PB96–0098-C04–04 y PAI TIC-137.
†Financiado parcialemente por el proyecto DGES PB96–1345 y PAI TIC-137.

ternaria $t : R^3 \to R$ (donde R es un conjunto en correspondencia biyectiva con todas las rectas que pasan por un punto, salvo una), ciertas propiedades de la aplicación, definiendo el concepto de *anillo ternario plano*. Por ejemplo, en el plano real, la ecuación $y = t(x, m, b)$ representa la recta de $y = mx + b$.

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])[1]. 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 [Klucký 95]:

Definición 1 *Una estructura* $\langle M, t, ^*, \{0_L, 0_R\}\rangle$ *es un anillo plano ternario natural (NPTR) si verifica las siguientes condiciones:*

(A) $\forall x, y, m \exists! b[t(x, m, b) = y]$.

(B) $\forall m, m', b, b'[m \neq m' \to \exists! x(t(x, m, b) = t(x, m', b'))]$.

(C) $\forall x, x', y, y'[x \neq x' \to \exists! m, b(t(x, m, b) = y \wedge t(x', m, b) = y']$.

(D) $\forall m, b[t(0_L, m, b^*) = b] \wedge \forall x, b[t(x, 0_R, b^*) = b]$.

(E) *La función* $(.)^*$ *es biyectiva.*

3 Preprocesamiento

Para utilizar un demostrador automático de manera eficiente, es necesario reescribir los axiomas que definen a los NPTR. 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 $t, (.)^*$ 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.

El objetivo es encontrar una nueva axiomatización, simplificando la complejidad lógica de los axiomas, tanto de los cuantificadores como de la estructura interna de la fórmula. Con respecto a los cuantificadores se utilizan las denominadas *funciones de Skolem*. Intuitivamente, una función de Skolem representa la dependencia funcional de una variable con respecto a otras, y se fundamenta en el siguiente resultado clásico:

Teorema 2 *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*

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

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

[1] 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.

Aplicando adecuadamente el resultado a los axiomas de una teoría T, se obtiene una extensión conservativa de T cuyos axiomas son fórmulas sin cuantificadores existenciales. Una vez eliminados los cuantificadores existenciales, 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 aplicando los dos procedimientos es la siguiente teoría $NPTR_0$, formada por las siguientes disyunciones (llamadas cláusulas)[2]:

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

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

4 Mecanismos de inferencia

El siguiente paso para trabajar con la teoría es analizar la estructura de sus axiomas. 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, las reglas que utilizaremos son las siguientes:

- Resolución binaria:

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

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

- Hiperresolución:

$$\frac{M_1\ \cdots\ M_k \qquad -A_1|\ldots|-A_k|B_1|\ldots|B_m}{(B_1|\ldots|B_m)\sigma}$$

  (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.

[2]Las cláusulas de tipo E son de *existencia de solución*, las de tipo U son de *unicidad*, el símbolo | denota la disyunción y $-$ es el símbolo de negación.

- 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:

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

  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.

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 (la idea se basa en que 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 o, en su caso, las variantes de dichas reglas y la partición del conjunto de fórmulas original. Hay que tener en cuenta que la estrategia del conjunto soporte no es completa (no cualquier partición del conjunto de entrada lleva a una inconsistencia). De las cláusulas así generadas, OTTER utiliza diversos mecanismos para descartar las que son redundantes, es decir, son un caso particular de otras.

6 Teoremas demostrados

En esta sección comentamos las pruebas obtenidas con el demostrador. Una primera prueba del teorema la hemos obtenido mediante el modo `auto2` de OTTER, 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 experimento ha sido realizado utilizando la última versión UNIX/LINUX del programa, en un PC (AMD K6II–450, 64 Mb RAM). Por cuestiones de espacio, sólo presentamos algunas características de dichas pruebas.

6.1 Unicidad de 0_L, 0_R

Presentamos en esta sección las pruebas obtenidas de la unicidad de los elementos 0_L y 0_R en un NPTR.

Teorema 3 *Los elementos 0_L y 0_R son únicos con esa propiedad.*

Demostración:
Unicidad de 0_L: Encontrada por OTTER en 0.02 segundos en modo `auto2`, generándose 48 cláusulas en una prueba de longitud 7. Reduciendo el conjunto de reglas al formado por ur-resolución y las variantes de paramodulación y demodulación (para-into y back-demod, respectivamente), y utilizando únicamente los axiomas B_U y D_1, obtenemos en 0.01 segundos una prueba más simple en la que se generan 20 cláusulas.
Unicidad de 0_R: Encontrada por OTTER, en una primera aproximación , en 0.22 segundos, generándose 297 cláusulas.Utilizándose las reglas ur-resolución y para-into, y con los axiomas B_U y D_2, conseguimos en 0.02 segundos una prueba en la que se generan tan solo 8 cláusulas. ∎

7 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 la que hemos demostrado las propiedades fundamentales presentadas en [Klucký 95]. Por razones de espacio no presentamos las pruebas obtenida. En su lugar mostramos una tabla en la que aparecen los datos de las pruebas obtenidas por OTTER en tres versiones. La primera, en modo `auto2`, disponiendo de todos los axiomas iniciales. En segunto lugar, en modo `auto2` reduciendo el conjunto de axiomas utilizables a los implicados en esa primera prueba. Por último, los datos de la prueba obtenida seleccionando las reglas y los axiomas en base a esas dos primeras aproximaciones:

Proposición 4 *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/Seg.	Cl. generadas	Long. prueba
(1), unicidad	0.54/0.1/0.06	604/327/590	14/14/7
(1), existencia	88.54/0.03/0.02	7127/143/145	20/9/9
(2), unicidad	0.84/0.23/0.01	675/355/55	14/14/6
(2), existencia	17.15/0.01/0.01	4114/23/9	40/5/4
(3), $\Longrightarrow$	0.99/0.08/0.02	753/249/29	14/10/6
(3), $\Longleftarrow$	15.79/0.01/0.00	3735/9/14	15/7/2

8 Propiedades de la suma

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

```
(e(ol) = or)
(X = ol) | (prod(X,e(X)) = X)
(X = ol) | (prod(X,Y) != X) | (e(X) = Y)
```

Así, definimos una nueva operación, $+$, como

$$a + b = t(a, e(a), b^*)$$

Proposición 5 *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/Seg.	Cl. generadas	Long. prueba
(1) $a + 0_L = a$	98.05/0.02/0.01	5795/22/9	16/6/5
(1) $(0_L + a) = a$	2.23/0.01/0.00	1161/11/3	2/2/1
(2), existencia	0.42/0.01/0.02	562/33/43	6/6/7
(2), unicidad	45.92/0.03/0.01	4837/89/75	10/6/4

8.1 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 4,

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

puede ser muy útil 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 es posible. Demostrar la existencia de un objeto, no es suficiente para que tal objeto se pueda expresar de manera simple en el lenguaje con el que trabajamos. En el caso que nos ocupa (y esta es otra de las ventajas de preprocesar la teoría, obteniendo una extensión conservativa formada por cláusulas) esta propiedad es cierta, utilizando el *teorema de Herbrand*. Como consecuencia, se tiene que:

Corolario 6 [Alonso et al. 00] *Existe un L_0–término $\mathbf{t}(u, v)$ tal que*

$$NPTR_0 \vdash u \neq 0_R \to \mathbf{t}(u, v) \cdot u = v$$

Por tanto, en nuestro caso, tiene sentido intentar obtener una expresión de este tipo. Nótese que esta fundamentación de la obtención de respuestas es una generalización del concepto de respuesta utilizada en programación lógica (por ejemplo en PROLOG), véase [Kunen 96]. De hecho, OTTER tiene implementada una función, $\$ans(\mathtt{X})$, que devuelve el valor obtenido para la variable `X`. En nuestro caso, 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 0.01 seg.) nos devuelve la solución $ans(tb(or, asterisco(b), a, asterisco(ol)))$, es decir,

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

Esta 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 a una simplificada. Como ya hemos justificado anteriormente, cualquier solución denota el mismo elemento, pues es único. 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

$$t(x, 0_R, b^*) = t(x, a, (0_R)^*)$$

8.2 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 $x, m, b \in M$

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

El teorema afirma 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 $(M, t_2, ^\times, \alpha^{-1}(0_L), \beta^{-1}(0_R))$ es un NPTR. Para encontrar tal función, razonamos como en la subsección anterior, y tomamos como conjunto soporte las siguientes fórmulas:

```
delta(t2(X,M,B)) = t(alfa(X),beta(M),gamma(B)).
(inversa-delta(delta(X)) = X).
(delta(inversa-delta(X)) = X).
(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(ol2) = ol).
(beta(or2) = or).
(all X ((all M (t2(ol2 , M , X) = b)) -> $ans(X))).
```

La última de las cuales exige la respuesta `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, $x^\times = \gamma^{-1}(\delta(x)^*)$ es la aplicación pedida.

9 Conclusiones

En este artículo hemos mostrado el uso fundamentado de un demostrador automático como asistente para la investigación en Matemáticas, concretamente para la investigación en anillos ternarios. Asímismo hemos hecho notar la utilidad del diseño de estrategias para eliminar en el proceso de deducción la información redundante, como es el caso de reducir el conjunto de axiomas en juego.

Análogamente es posible conducir la búsqueda de prueba seleccionando las reglas más idóneas o utilizando variantes de las mismas que ahorran tiempo y ayudan a simplificar las demostraciones e incluso la expresión de las respuestas (ver corolario 6).

En ocasiones ha sido de utilidad ampliar el conjunto soporte, añadiendo algunas cláusulas a la negación del teorema a probar, como en el caso de la búsqueda de la función relativa a isotopías en 8.2.

Es necesario incidir, a la vista de los datos comparativos ofrecidos por las tablas de las pruebas, en el hecho de que el modo `auto2` es mejorable, en general, por un usuario. Pero no en todos los casos, como en el de la proposición 6.2 (existencia).

Por último, señalemos que hemos utilizado el método de obtención de respuestas para probar la existencia de un objeto y conocer a su vez una expresión de dicho objeto, pudiendo además comprobar fácilmente que la respuesta dada por el demostrador es correcta.

Como línea futura de investigación en este campo cabría insistir en el diseño y perfeccionamiento de estrategias para la obtención de pruebas mediante procesos cada vez más cortos en tiempo y longitud, en la obtención de respuestas con expresiones sencillas, así como en el diseño de nuevas estrategias para la demostración que utilicen información conocida del modelo matemático de estudio (en este caso, información geométrica).

Referencias

[Alonso et al. 00] Alonso-Jiménez, J.A.; Borrego-Díaz, J.; Chávez-González, A.:*Deducción automática en anillos ternarios: Algunos métodos de procesamiento del conocimiento matemático.* Aparecerá en Encuentro de Matemáticos Andaluces, 2000.

[Blumenthal 61] Blumenthal, L.M.: A modern view of Geometry. Freeman, 1961.

[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 Reasonning 19 (3), 263–276 (1997).

[Wos et al. 65] Wos, L.; Robinson, G.; Carson, D.: *Effinciency and completeness of the set of support strategy in theorem proving.* J. ACM 12 (4), 536–541 (1961).