

Descubrimiento Automático en un problema Centenario

por

Rafael Losada y Tomás Recio*

1. EL CONTEXTO

Desde hace unos años se va abriendo camino, lentamente, una línea de trabajo en la que confluyen la teoría de la demostración automática, el álgebra polinomial computacional, el software de geometría dinámica y la educación matemática. El objetivo genérico de esa conjunción de esfuerzos es lograr programas de geometría dinámica (como *Cabri*¹ o *GeoGebra*², por citar dos de los posiblemente más conocidos por los lectores de *La Gaceta*) capaces de

- obtener, automáticamente, una demostración sobre la verdad o falsedad de cualquier cuestión planteada por el usuario en torno a una construcción realizada por el mismo con el programa [2];
- la aportación, en su caso, de información de interés para facilitar al usuario el descubrimiento de la demostración por sus propios medios (por ejemplo, a través de la generación automática de figuras auxiliares, o mediante la propuesta, corroboración o refutación de conjeturas que el usuario pueda elaborar como pasos intermedios hacia la demostración buscada, [15], [14], [8]);
- la deducción automática de propiedades de determinados elementos de una figura (por ejemplo, dado un triángulo, si se introdujera en el programa la noción de área del mismo y las longitudes de sus lados y se seleccionasen estos datos como objeto de deducción, el programa debería hallar automáticamente la fórmula de Herón [3]);
- el descubrimiento automático de teoremas, en el sentido de proporcionar automáticamente la hipótesis o las hipótesis necesarias y suficientes para que cierto enunciado conjetural (generalmente falso sobre la figura construida) sea cierto [9], [4]. Por ejemplo, el usuario afirma que en todo triángulo existe un vértice en el que coinciden las longitudes de los dos segmentos bisectores (determinados por la bisectrices internas y externas desde el vértice hasta su intersección con el lado opuesto) y el programa responde que no es cierto en general, pero que lo es si y solo si el triángulo tiene tal o cual propiedad [7].

*Parcialmente financiado por el proyecto MTM2008-04699-C03-03.

¹<http://www.cabri.com/es/>

²<http://geogebra.es/>

Existen ya algunos programas que, de manera experimental, incorporan algunas de estas características. Entre ellos, por su carácter pionero, debemos citar *GDI*, las iniciales de *Geometría Dinámica Inteligente*, un programa pionero desarrollado hace casi 10 años por los profesores Francisco Botana y José Luis Valcarce [1].

Es importante mencionar, incluso en esta breve introducción, que esta línea de trabajo se ha visto reforzada por la amplia difusión de *GeoGebra* y por su apuesta por la unión, en un mismo programa, de la geometría dinámica y el cálculo simbólico, las dos herramientas imprescindibles para alcanzar los objetivos arriba señalados. Ahora, la consecución de los mismos es una tarea conjunta de investigación (sobre los algoritmos algebraicos implicados), de innovación (diseñando escenarios para el aprovechamiento, en el ámbito educativo o en otros ámbitos, de estas técnicas [12]), y de desarrollo e implementación (en programas tales como *GeoGebra*).

2. EL PROBLEMA DEL CENTENARIO

Hace unos meses, en esta misma revista [10], se ha planteado el siguiente *Problema del Centenario*: Dado un triángulo cualquiera, se construye en su interior otro triángulo, que llamaremos triángulo *bisector*, formado por los pies de la bisectrices de cada ángulo en el lado opuesto al mismo. Se toma ahora un punto cualquiera del perímetro de dicho triángulo bisector y se consideran las distancias d_1, d_2, d_3 de dicho punto a los lados del triángulo dado. El problema nos propone probar, entonces, que una de estas distancias es siempre igual a la suma de las otras dos, esto es: $d_1 - d_2 - d_3 = 0$ ó $d_2 - d_1 - d_3 = 0$ ó $d_3 - d_1 - d_2 = 0$.

Es preciso indicar que el enunciado original y la figura que lo acompañaba, así como la curiosa historia de su origen y una solución «tradicional» a este problema aparecen en distintos números anteriores de *La Gaceta* [10], [11], por lo que no consideramos oportuno entrar aquí en más detalles.

Ahora bien, la sencillez de su enunciado y su planteamiento en un contexto «Centenario» hacen a este problema un candidato natural para ser abordado mediante técnicas automáticas. Es preciso señalar que lo que sigue *no* ha sido efectuado de forma totalmente automática, sino *simulando* (con intervención humana, de los autores de este artículo) en varias ocasiones la interrelación entre un hipotético programa de geometría dinámica y un programa de álgebra computacional, como Maple, con el que se han efectuado —automáticamente, eso sí— los distintos cálculos.

2.1. PLANTEAMIENTO

Para complicar un poco más las cosas abordamos el problema desde la perspectiva de la deducción automática: suponemos que hemos hecho la construcción del triángulo bisector y que queremos *descubrir* qué relación verifican las distancias de un punto cualquiera de su perímetro a los lados del triángulo dado. Es decir, no suponemos como conocida la tesis del problema, sino que pedimos al ordenador que descubra la relación existente, si hay alguna, entre esas distancias.

Para ello (supuestamente, en un programa inteligente de geometría dinámica), y tal como se muestra en la figura 1, dibujaremos un triángulo *ABC*, construire-

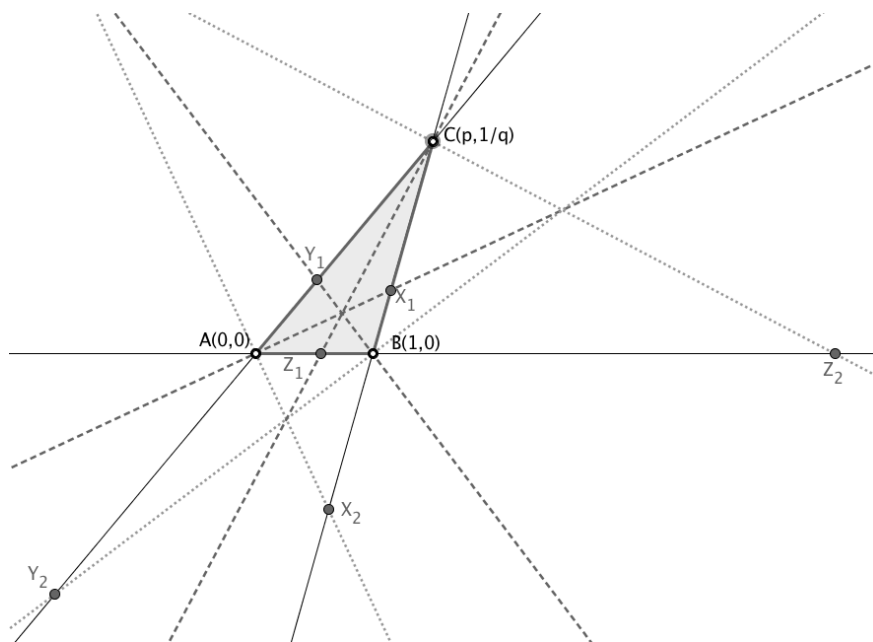


Figura 1: Planteamiento.

mos las bisectrices de cada vértice, trazaremos los puntos de intersección con el lado opuesto y tomaremos un punto arbitrario de su perímetro. Entonces haremos la construcción de la perpendicular desde ese punto a los tres lados y tomaremos los segmentos correspondientes, cuya longitud será cada una de las distancias sobre la cuales queremos tener un enunciado. Toda esta aproximación al problema es perfectamente realizable en un programa de geometría con capacidades de descubrimiento automático: *GDI*, en particular, las tiene ya integradas.

A fin de simplificar el número de variables, sin pérdida de generalidad, es habitual suponer que el vértice *A* tiene coordenadas $(0, 0)$ y el vértice *B* coordenadas $(1, 0)$. Esta opción también está disponible en *GDI*. Al vértice *C*, por el contrario, se le asignan coordenadas variables, qu , en este caso hemos considerado oportuno denominar $C = (p, 1/q)$, a fin de evitar —en cierto sentido— que la segunda coordenada de *C* se haga cero y el triángulo colapse.

Lo que sigue es mecánico y plenamente mecanizable (de hecho, *GDI* tiene ya implementada la creación automática de las ecuaciones asociadas a los distintos pasos de una construcción, algo que también será fácil de realizar con *GeoGebra*, por su especial vinculación con el álgebra). Con la notación introducida arriba, las longitudes de los lados $a = BC$, $b = AC$ y $c = AB$ quedan definidas por

$$a^2 = (p - 1)^2 + 1/q^2, \quad b^2 = p^2 + 1/q^2, \quad c = 1,$$

ya que los algoritmos más eficaces en este contexto son aquellos que operan con polinomios y es aconsejable no introducir raíces cuadradas (ver discusión en la sub-

sección siguiente). Las ecuaciones de las rectas BC , AC y AB son

$$BC : x - (p - 1)qy - 1 = 0, \quad AC : x - pqy = 0, \quad AB : y = 0.$$

Sea (x, y) un punto cualquiera. Los cuadrados de las distancias d_1, d_2, d_3 de ese punto a los lados serán entonces

$$d_1^2 = (x - (p - 1)qy - 1)^2 / (a^2 q^2), \quad d_2^2 = (x - pqy)^2 / (b^2 q^2), \quad d_3^2 = y^2,$$

que, en los cálculos posteriores, serán usados multiplicando por los denominadores (esto es, $(a^2 q^2)d_1^2 = (x - (p - 1)qy - 1)^2$, etc.). Por otra parte, los vectores $AB = (1, 0)$ y $(1/b) * AC = (1/b) * (p, 1/q)$ tienen la misma longitud (son ambos unitarios). Por tanto su suma nos indica la dirección de la bisectriz interior, que será $(1/b) * (p + b, 1/q)$, es decir, la dirección es $((p + b)q, 1)$. Por tanto, la bisectriz interior en A tiene por ecuación $-x + (p + b)qy = 0$, y su pie $X_1(x_1, x_2)$ quedará definido por su intersección con la recta BC , mediante el sistema de ecuaciones

$$-x_1 + (p + b)qx_2 = 0, \quad x_1 - (p - 1)qx_2 - 1 = 0,$$

que es equivalente a

$$(1 + b)x_1 - (p + b) = 0, \quad (1 + b)qx_2 - 1 = 0.$$

De modo similar construimos las otras bisectrices y los correspondientes pies, de coordenadas $Y_1(y_1, y_2)$ para B y $Z_1(z_1, 0)$ para C , resultando las siguientes ecuaciones:

$$(1 + a)y_1 - p = 0, \quad (1 + a)qy_2 - 1 = 0, \quad (a + b)(z_1 - p) + (ap + bp - b) = 0.$$

Ahora consideramos uno de los lados del triángulo definido por los pies de las bisectrices, por ejemplo, el lado X_1Y_1 , y un punto (x, y) arbitrario en el mismo, verificando, por tanto, la ecuación

$$(y_1 - x_1)(y - x_2) - (y_2 - x_2)(x - x_1) = 0.$$

El objetivo ahora es deducir alguna relación entre $\{d_1, d_2, d_3\}$ que se verifique como consecuencia de las ecuaciones $\{(y_1 - x_1)(y - x_2) - (y_2 - x_2)(x - x_1) = 0, (1 + a)y_1 - p = 0, (1 + a)qy_2 - 1 = 0, (1 + b)x_1 - (p + b) = 0, (1 + b)qx_2 - 1 = 0, a^2 q^2 d_1^2 - (x - (p - 1)qy - 1)^2 = 0, b^2 q^2 d_2^2 - (x - pqy)^2 = 0, d_3^2 - y^2 = 0\}$. Y también se trata de hallar las relaciones entre esas distancias para puntos en los lados X_1Z_1 y Z_1Y_1 , modificando convenientemente las ecuaciones del sistema anterior para recoger la ecuación de la recta definida por el lado correspondiente y la definición de z_1 .

Desde un punto de vista algorítmico, la deducción buscada consiste en *eliminar* todas las variables del sistema de ecuaciones, excepto $\{d_1, d_2, d_3\}$. El resultado, si hay teorema, será una o varias expresiones en estas variables, que se verificarán para todos los puntos de cada uno de los lados del triángulo bisector, entre las cuales, esperamos, descubriremos la propiedad enunciada en [10], esto es, que una de las distancias es igual a la suma de las otras dos. El programa de álgebra computacional *Maple* (y otros más especializados, como *CoCoA* o *Singular*) tienen comandos específicos para lograr esta eliminación. ¿Fin del problema?

2.2. REFLEXIÓN

Llegados a este punto es preciso decir que tras varios intentos (decenas de horas de cálculo y centenares de megas de memoria) sobre un ordenador personal —como el que, idealmente, usarían los destinatarios de los futuros programas de geometría dinámica inteligente— el proceso de eliminación no termina. ¿Qué ocurre?

Una primera observación es que nuestra versión algebraica de la construcción de las bisectrices realmente incluye, simultáneamente, tanto las bisectrices interiores como las exteriores (perpendiculares a las anteriores) en cada uno de los vértices, porque los valores de a, b no están definidos sino salvo signo. Y las distancias a los lados que hemos considerado están, también, definidas mediante cuadrados, sin especificar el signo.

El problema de manejar soluciones con signo (o desigualdades) es propio de la geometría y álgebra computacional real y existen, en ese contexto, excelentes aproximaciones a la demostración automática de teoremas geométricos (tanto teóricas [5], como implementadas en un software específico [13]). Pero su capacidad para resolver problemas en un contexto educativo es mucho menor, y la complejidad de su implementación (y conexión con un programa de geometría dinámica) es mucho mayor. Digamos que se trata, en el actual estado de desarrollo, de un producto «avanzado», frente al objetivo de popularización de nuestra propuesta.

Así pues, si tenemos que optar por trabajar en un entorno «sin signos», donde los algoritmos algebraicos son más sencillos y eficaces, el problema que estamos formulando es, realmente, el descubrimiento de una propiedad de las distancias a los lados de un punto que esté en el perímetro de *cualquiera* de los triángulos bisectores, los determinados por los *seis* pies de las bisectrices (tres externas y tres internas). ¿Existe realmente una propiedad de $\{d_1, d_2, d_3\}$ que se cumpla en cualquiera de esos triángulos?

Una interfaz amigable de ese futuro programa de geometría dinámica inteligente debería avisar al usuario de que la construcción que está realizando incluye, inevitablemente, otros elementos que tal vez no haya considerado, y debería ofrecer esa construcción completa asociada a la realizada por el usuario. En este punto, el usuario tal vez se pregunte cuántos triángulos bisectores vienen dados por los seis pies de las bisectrices (dos por cada vértice, tales como X_1, X_2 para el vértice A ; Y_1, Y_2 para el vértice B ; Z_1, Z_2 para el vértice C , véase la figura 1). Teóricamente deberían presentarse ocho triángulos: $X_1Y_1Z_1, X_1Y_1Z_2, X_1Y_2Z_1, X_1Y_2Z_2, X_2Y_1Z_1, X_2Y_1Z_2, X_2Y_2Z_1, X_2Y_2Z_2$, pero una simple inspección a la figura 2 nos hace dudar de esa afirmación.

En efecto, en la figura «parece» que siempre ocurre que dos pies de sendas bisectrices internas están alineados con el pie de la bisectriz externa del vértice restante y, además, los tres pies de las bisectrices externas también están alineados.

2.3. UN TEOREMA AUXILIAR

Tal vez el futuro usuario de un programa de geometría dinámica inteligente, llegado a este punto, quiera corroborar esta intuición, mostrando la alineación de

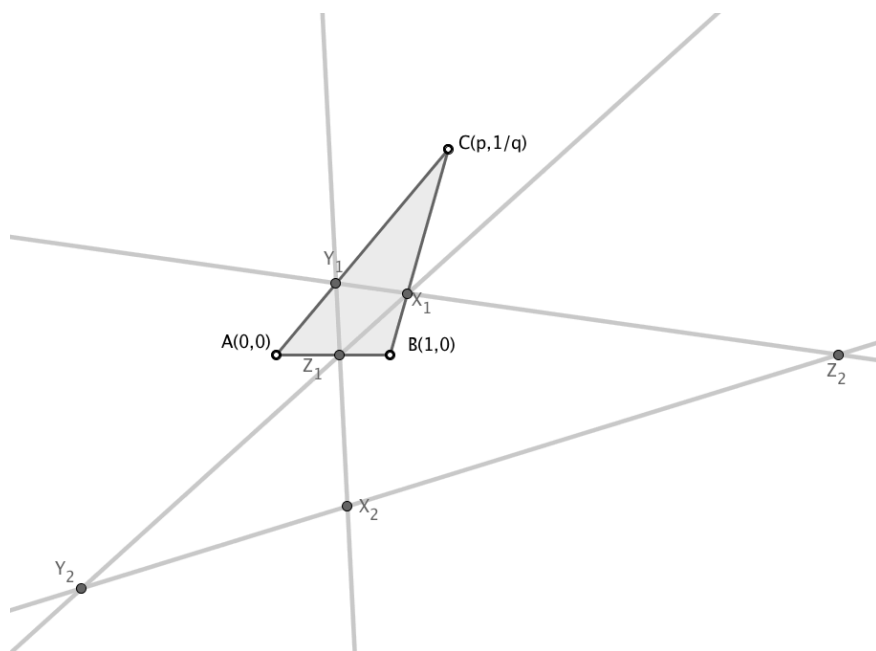


Figura 2: Alineación de pies de las bisectrices.

las ternas de puntos (X_1, Y_1, Z_2) , (X_1, Y_2, Z_1) , (X_2, Y_1, Z_1) , (X_2, Y_2, Z_2) . Recordemos que las ecuaciones que definen cada uno de estos puntos ya han sido introducidas (en efecto, no hay forma de distinguir X_1 y X_2 , Y_1 e Y_2 , Z_1 y Z_2 , los pies de las bisectrices internas y externas, respectivamente, salvo por el signo de a, b). Consideremos, por ejemplo, el caso de la terna (X_1, Y_1, Z_2) , cuyas coordenadas son $X_1(x_1, x_2)$, $Y_1(y_1, y_2)$ y $Z_2(z_3, 0)$.

Así pues, introducimos en *Maple* el conjunto de ecuaciones N (en las que, por la sintaxis de Maple, hemos eliminado el término $= 0$) que describen estos pies de las bisectrices, asignando un signo a las distancias a, b , según los casos, y luego eliminamos todas las variables menos $\{x_1, x_2, y_1, y_2, z_3\}$, para verificar que la expresión algebraica de la alineación de esos tres puntos es una combinación de las ecuaciones dadas.

```
> with(PolynomialIdeals): with(Groebner):
> N := <(p-1)^2*q^2-a^2*q^2+1, p^2*q^2-b^2*q^2+1, (1+b)*x_1-(p+b),
      (1+b)*q*x_2-1, (1+a)*y_1-p, (1+a)*q*y_2-1,
      (a-b)*(z_3-p)+(a*p-b*p+b)>;
> H := EliminationIdeal(N, {x_1, x_2, y_1, y_2, z_3});
> op(4, H);
```

obteniendo, casi inmediatamente, que el polinomio número cuatro del resultado de la eliminación es

$$z_3 x_2 + x_1 y_2 - x_2 y_1 - z_3 y_2$$

que, igualado a cero, describe la alineación de (X_1, Y_1, Z_2) . La comprobación o demostración de la alineación de las otras ternas se obtiene análogamente.

2.4. FINALMENTE, LA SOLUCIÓN

Como consecuencia de esta alineación de ternas de pies de las bisectrices, es evidente que el teorema cuya tesis queremos descubrir debe proporcionar un enunciado sobre las distancias a los lados de un punto genérico (x, y) situado sobre alguna de las cuatro rectas X_1Y_1 , X_1Y_2 , X_2Y_1 y X_2Y_2 , puesto que son las únicas que aparecen como lados de los triángulos bisectores. Todas ellas vienen representadas por una única ecuación tal como $(y_1 - x_1)(y - x_2) - (y_2 - x_2)(x - x_1)$, puesto que no hay forma de distinguir, sin introducir signos, el punto $X_1(x_1, x_2)$ y el punto $X_2(x_3, x_4)$, etc.

Procedemos como anteriormente, introduciendo en *Maple* las ecuaciones (sin poner al final $= 0$) de los puntos, rectas y distancias y tratando de eliminar todas las variables menos las que involucran las distancias, para saber qué expresión se verifica —si existe alguna— en esas solas variables:

```
> with(PolynomialIdeals): with(Groebner):
> NN := <(y_1-x_1)*(y-x_2)-(y_2-x_2)*(x-x_1), (p-1)^2*q^2-a^2*q^2+1,
      p^2*q^2-b^2*q^2+1, (1+b)*x_1-(p+b), (1+b)*q*x_2-1, (1+a)*y_1-p,
      (1+a)*q*y_2-1, (x-(p-1)*q*y-1)^2-a^2*q^2*d_1^2,
      (x-p*q*y)^2-b^2*q^2*d_2^2, y^2-d_3^2>;
> HH := EliminationIdeal(NN,{d_1,d_2,d_3});
```

El resultado es decepcionante. Los cálculos son excesivos y el programa no termina.

Llegados a este punto, la interfaz de ese futuro programa inteligente para la demostración en geometría debería proporcionar al usuario la posibilidad de incluir algunas condiciones de «no degeneración» (de nuevo, algo bastante simple e implementado ya en *GDI*), tales como añadir, a las ecuaciones anteriores, otra que indique que a, b, q no han de ser nunca cero³ (pues, en otro caso, el triángulo degeneraría). La máquina traduce internamente este requerimiento añadiendo la ecuación $abqW - 1 = 0$, donde W es una variable auxiliar. Claramente, si $abqW - 1 = 0$, no podrán anularse ninguna de las variables a, b, q . Sin embargo, tampoco es posible terminar los cálculos, aun con esta ayuda del usuario. ¿Cómo seguir?

Una de las cosas que hemos aprendido, en el desarrollo de esta solución «automática» al problema 3 del Centenario, es la conveniencia de aumentar las posibilidades de diálogo usuario-máquina, o la de tener ya implementados —en casos en los que fracasa la aproximación más directa— algunos heurísticos tales como el que vamos a describir a continuación⁴. Obsérvese que las tres ecuaciones últimas de NN, esto es

$$(x - (p - 1)qy - 1)^2 - a^2q^2d_1^2 = 0, \quad (x - pqy)^2 - b^2q^2d_2^2 = 0, \quad y^2 - d_3^2 = 0,$$

³Estrictamente hablando, en nuestro caso alguna de estas condiciones se pueden deducir directamente de las ecuaciones utilizadas.

⁴Agradecemos al profesor J. Rafael Sendra, de la Universidad de Alcalá, su crucial ayuda en este punto y en la obtención de los cálculos que siguen.

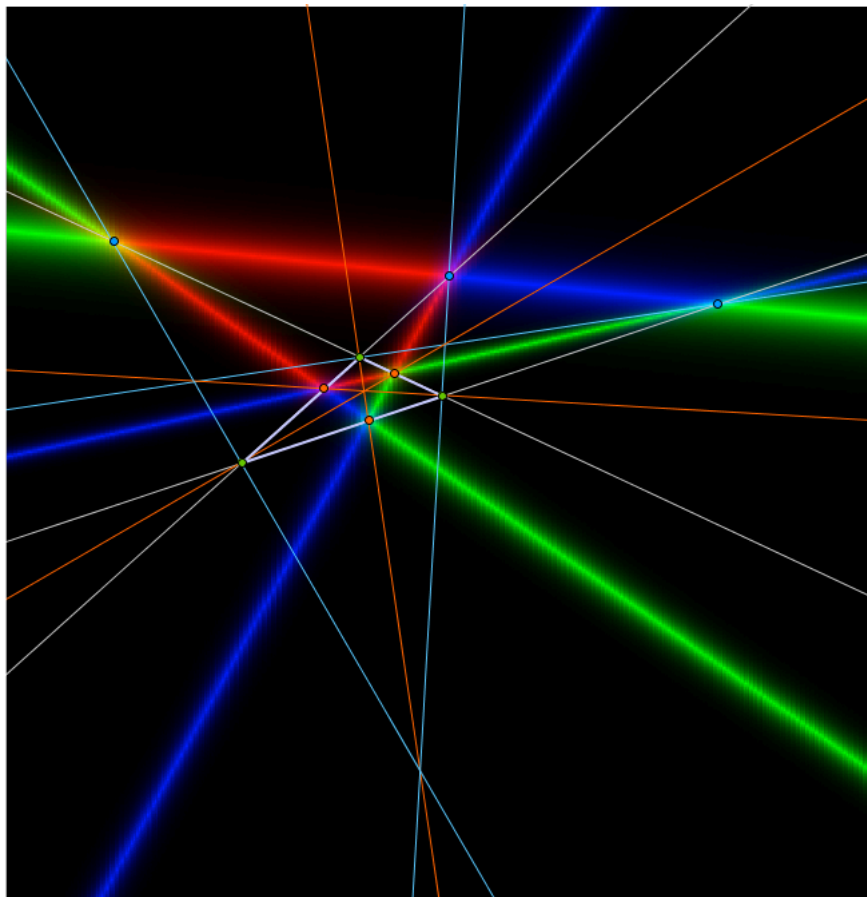


Figura 3: Triángulos bisectores.

son, cada una, diferencia de cuadrados y, por tanto, elementalmente factorizables. Si en vez del sistema NN escribimos los ocho sistemas NN_i , $i = 1, \dots, 8$, que resultan de reemplazar en NN cada una de las tres últimas ecuaciones por alguno de sus dos factores, cada solución de NN será solución de alguno de estos ocho sistemas, y recíprocamente. Así pues, si eliminamos en cada uno de los sistemas NN_i , $i = 1, \dots, 8$, todas las variables menos las variables $\{d_1, d_2, d_3\}$ y multiplicamos todos los resultados obtenidos, habremos obtenido una expresión en $\{d_1, d_2, d_3\}$ válida para el sistema NN .

La ventaja de los sistemas NN_i es, evidentemente, su mayor simplicidad, al tener ecuaciones más sencillas. Para maximizar nuestras posibilidades de éxito, además, hemos añadido en cada sistema la condición de no degeneración que hemos comentado arriba. El resultado obtenido ha sido el siguiente (el hecho de que aparezca

elevado al cuadrado es irrelevante):

$$(-d_2 - d_3 + d_1)^2(-d_2 + d_1 + d_3)^2(d_2 + d_1 - d_3)^2(d_2 + d_1 + d_3)^2 = 0,$$

es decir, que se verifica siempre que la suma —con signo— de las tres distancias ha de ser cero (incluyendo el caso de la suma de las tres, puesto que nuestra aproximación no distingue entre distancias positivas o negativas).

En la figura 3, obtenida sobre *GeoGebra* usando un procedimiento que se describe en [6], de color dinámico, se puede apreciar, en color más claro, el pequeño triángulo original y, en otros tonos, las bisectrices interiores y exteriores de cada vértice y las cuatro rectas que forman los lados de los cuatro triángulos bisectores reales. En la versión en color de esta figura (o si se usa la aplicación de *GeoGebra* que se describe en [6]) se pueden diferenciar (en azul, rojo o verde) aquellos segmentos del perímetro de los lados del triángulo bisector en los que se verifica una de estas condiciones sobre las distancias.

En resumen, hemos descubierto y demostrado —¿automáticamente?— la siguiente generalización del problema 3 del Centenario:

TEOREMA. *Dado un triángulo cualquiera, se consideran todos sus triángulos bisectores, formados por los pies de las bisectrices internas o externas, un pie por cada vértice. Entonces, dado un punto del perímetro de cualquiera de estos triángulos bisectores, se verifica que una de las distancias del punto a los lados del triángulo dado es igual a la suma de las otras dos.*

Quede para el infatigable lector la tarea de demostrar este enunciado por el procedimiento convencional...

REFERENCIAS

- [1] F. BOTANA Y J.L. VALCARCE, A Dynamic-Symbolic Interface for Geometric Theorem Discovery. *Computers & Education*, **Vol. 38**, (1-3), (2002), Págs. 21–35.
- [2] SHANG-CHING CHOU, *Mechanical Geometry Theorem Proving*. D. Reidel Pub. Co., (1988).
- [3] SHANG-CHING CHOU Y XIAO-SHAN GAO, Mechanical Formula Derivation in Elementary Geometries. *Proceedings International Symposium on Symbolic and Algebraic Computation (ISSAC)*, ACM Press, New York, (1990), Págs. 265–270.
- [4] G. DALZOTTO Y T. RECIO, On Protocols for the Automatic Discovery of Theorems in Elementary Geometry. *Journal of Automated Reasoning*, **Vol. 43**, (2009), Págs. 203–236.
- [5] A. DOLZMANN, T. STURM Y W. WEISPFENNING, A New Approach for Automatic Theorem Proving in Real Geometry. *Journal of Automated Reasoning*, **Vol. 21**, Issue 3, (1998), Págs. 357–380.
- [6] R. LOSADA, Propiedad de color dinámico en *GeoGebra*, http://geogebra.es/color_dinamico/color_dinamico.html

- [7] R. LOSADA, T. RECIO Y J.L. VALCARCE, Equal Bisectors at a Vertex of a Triangle, *Lecture Notes in Computer Science*, **6785**, (2011), Págs. 328–341.
- [8] N. MATSUDA Y K. VANLEHN, GRAMY: A Geometry Theorem Prover Capable of Construction. *Journal of Automated Reasoning*, **Vol. 32, Issue 1**, (2004), Págs. 3–33.
- [9] T. RECIO Y P. VÉLEZ, Automatic Discovery of Theorems in Elementary Geometry. *Journal of Automated Reasoning*, **Vol. 23**, (1999), Págs. 63–82.
- [10] REDACCIÓN DE LA SECCIÓN DE PROBLEMAS Y SOLUCIONES, Los Problemas del Centenario. *La Gaceta de la RSME*, **Vol. 13, No. 4**, (2010), Págs. 723–724.
- [11] REDACCIÓN DE LA SECCIÓN DE PROBLEMAS Y SOLUCIONES, Los Problemas del Centenario: Resolución del concurso de la RSME, con motivo de su centenario, para participantes en el «XII Encuentro Nacional de Estudiantes de Matemáticas». *La Gaceta de la RSME*, **Vol. 14, No. 3**, (2011), Págs. 519–524.
- [12] R. RICHARD, J. FORTUNY, M. GAGNON, N. LEDUC, E. PUERTAS Y M. TESSIER-BAILLARGEON, Didactic and Theoretical-based Perspectives in the Experimental Development of an Intelligent Tutorial System for the Learning of geometry. *ZDM, The International Journal on Mathematics Education*, **Online**, (11 April, 2011), Págs. 1–15.
- [13] THE REDLOG HOME PAGE, <http://redlog.dolzmann.de/>
- [14] R. SOMMER Y G. NUCKOLS, A Proof Environment for Teaching Mathematics. *Journal of Automated Reasoning*, **Vol. 32, Issue 3**, (2004), Págs. 227–258.
- [15] ZHENG YE, SHANG-CHING CHOU Y XIAO-SHAN GAO, Visually Dynamic Presentation of Proofs in Plane Geometry I, II. *Journal of Automated Reasoning*, **Vol. 45, Issue 3**, (2010), Págs. 213–241 y 243–266.

RAFAEL LOSADA, IES DE PRAVIA, ASTURIAS

Correo electrónico: rafaell1@educastur.princast.es

Página web: <http://www.iespravia.com/rafa/rafa.htm>

TOMÁS RECIO, DEPARTAMENTO DE MATEMÁTICAS, ESTADÍSTICA Y COMPUTACIÓN, UNIVERSIDAD DE CANTABRIA, SANTANDER

Correo electrónico: tomas.recio@unican.es

Página web: <http://www.recio.tk>