CÁLCULO SIMBÓLICO, CÁLCULO FORMAL, ÁLGEBRA COMPUTACIONAL: QUÉ ES Y PARA QUÉ SIRVE.

Tomás Recio

96

Ideas sobre el álgebra computacional

Cálculo simbólico, cálculo formal, álgebra computacional... son distintos nombres para señalar una misma cosa. La mayor o menor difusión de cada una de estas denominaciones responde al origen de la misma (del inglés Computer Algebra o Symbolic Computation, del francés Calcul Formel). Según una definición ya clásica de R. Loos, "el álgebra computacional es aquella parte de las ciencias de la computación que diseña, analiza, implementa y aplica algoritmos algebraicos". Es cuestionable, en esta definición, que el álgebra computacional se considere una parte de las ciencias de la computación, posiblemente lo era más en la época en la que se acuñó esta descripción (principios de los ochenta). En todo caso, es evidente que su autor hace referencia a *algoritmos algebraicos* como la tarea característica del álgebra computacional y, naturalmente, esto establece una conexión directa con las matemáticas.

Puede que el lector piense, con toda razón, que esa definición es casi circular. Al fin y al cabo, suena casi igual decir álgebra computacional o algoritmos algebraicos. Así pues, ¿qué debe entenderse por algoritmos algebraicos? Si alguien preguntase por un ejemplo paradigmático y sencillo de algoritmo algebraico, una posible respuesta sería el algoritmo de Euclides de cálculo del máximo común divisor de números o polinomios. En definitiva, los algoritmos algebraicos del álgebra computacional son los procesos constructivos de un álgebra con símbolos... De ahí que al álgebra computacional también se la denomine cálculo simbólico.

Así el cálculo simbólico trabaja muchas veces en contextos algorítmicos alejados del álgebra tradicional. Por ejemplo, cuando en un programa informático como Maple o Mathematica calculamos la integral de una función racional, el algoritmo que está detrás es, por definición, un algoritmo de álgebra computacional... o cálculo simbólico, aunque las integrales, tradicionalmente, no sean álgebra. O cuando calculamos con esos programas un límite o cuando sumamos una serie... ha tenido que hacerse mucha álgebra computacional para obtener el resultado que aparece en la pantalla.

En la figura aparece una sesión típica (pero muy sencilla) de Maple, un popular programa de álgebra computacional. En el primer párrafo se solicita al programa que calcule la integral de una función racional. La respuesta muestra una función de la variable x, con dos logaritmos neperianos (simbolizados por las letras *ln*) y un arco tangente. En el segundo párrafo tratamos de hallar la diferencial del output del párrafo anterior (el símbolo % hace siempre refe-

```
> int ((x^3+x-1)/(x^4-1), x);
```

$$\frac{1}{4}\ln(x-1)+\frac{3}{4}\ln(x+1)+\frac{1}{2}\arctan(x)$$

```
> diff(%,x);
```

$$\frac{1}{4}\frac{1}{x-1}+\frac{3}{4}\frac{1}{x+1}+\frac{1}{2}\frac{1}{1+x^2}$$

```
> simplify(%);
```

$$\frac{x^3+x-1}{(x-1)(x+1)(1+x^2)}$$

```
> simplify((x^3+x-1)/(x^4-1)-%);
```

$$0$$

```
> solve(denom((x^3+x-1)/(x^4-1)),x);
```

$$1, -1, I, -I$$

```
> limit(((x^3+x-1)/(x^4-1)), x=infinity);
```

$$0$$

rencia, en el lenguaje de Maple, al último output), con respecto a la variable x. El resultado es una suma de tres fracciones. Naturalmente, si simplificamos esta suma (tercer párrafo), el resultado es una fracción cuyo denominador tiene grado 4. Para estar seguros de que la derivada de la integral es la función
96 dada, en el cuarto párrafo se pide a Maple que simplifique la diferencia entre la función dada y el resultado de derivar la integral obtenida. El resultado es cero, como cabía esperar. En el quinto párrafo se muestra otra de las posibilidades de Maple: se le pide que resuelva la ecuación dada por el denominador (*denom*) de la fracción que estamos usando en este ejemplo. La respuesta incluye la unidad imaginaria (que Maple escribe con mayúsculas) y su opuesta. En el último párrafo Maple calcula el límite de esa fracción cuando x tiende a infinito.

Una perspectiva histórica

La historia del cálculo simbólico es reciente o antigua, según se mire. La idea de manipular símbolos matemáticos de forma automática es antiquísima: cualquier libro de historia hace referencia a los sueños de Leibniz, por ejemplo, de construir un lenguaje formalizado con el que una máquina de cálculo pudiera deducir todos los teoremas y resultados apetecidos. El conde de Stanhope, en el siglo XVIII, trata de desarrollar una máquina para resolver silogismos, que en definitiva, es una máquina de cálculo simbólico.

En muchos sentidos, detrás del invento de los ordenadores actuales está C. Babbage, como es bien sabido. Pues el mismo Charles Babbage escribe, en julio de 1836, en su libro de notas: “Hoy tuve por primera vez una concepción