

Joshua Tauber

Artículo de respuesta a las lecturas de la semana nº 5

Mi área de investigación son los métodos formales para la verificación de la computación distribuida. Uno de los principales objetivos de nuestro grupo de investigación (especialmente de mi programa de investigación) es incrementar la automatización del razonamiento sobre la corrección de diseños de sistemas distribuidos y sus implementaciones.

Una de las principales herramientas que utilizamos es el Larch Prover (LP). LP es un asistente de pruebas automatizado. En la taxonomía de Donald MacKenzie, LP es un demostrador interactivo que genera pruebas distintas a las generadas por el hombre. La producción de una “prueba” con LP consiste generalmente en un proceso interactivo muy parecido al descrito por Boyer. El hombre formula una conjetura que LP posteriormente trata de demostrar. LP aplicará una o varias reglas disponibles con el fin de generar subobjetivos (supuestamente más simples) para demostrar. Lo más probable es que LP se quede atascado en algún momento del proceso y que necesite orientación para continuar. Por ejemplo, el usuario podría tener que proponer un valor que creará una instancia en un cuantificador existencial con el fin de descargar una ramificación del análisis del caso. El resultado al final de una interacción con éxito, es una afirmación realizada por LP en la que se ha demostrado la conjetura deseada.

El aspecto interesante es analizar lo que queda de este proceso. El usuario LP dispone ahora de un “comando” para demostrar el teorema. El script consta de la serie de comandos que el usuario introduce para hacer que LP acepte que la conjetura es verdadera. Es posible que las entradas del script introduzcan algunos pasos obvios que cualquier matemático reconocerá (al menos una vez que estén descodificadas) como "utilizar inducción sobre A" o "crear una instancia en B como D+E". Por otro lado, el script puede incorporar comandos que solicitan que LP lleve a cabo mayores operaciones como "utilizar pares críticos" (una operación relacionada con la resolución). Más interesante, sin embargo, es lo que no está allí. Lo que no está allí es el texto real de lo que se reconocería como prueba formal. Esto es, el script de prueba consta únicamente de las pistas (comandos) que deberá facilitar a LP cuando éste se quede atascado.

Para ser justos, se podría pedir a LP que imprimiese la mayoría de los pasos intermedios rigurosamente detallados. En la práctica, sin embargo, nadie lo hace. ¿Por qué? Por tres razones.

En primer lugar, los detalles de las pruebas formales no son en realidad muy interesantes. Los pasos exactos no cumplen una función explicativa. Esta es exactamente la misma razón por la que se han publicado las pruebas formales de textos casi incompletos. Los matemáticos se preocupan por la realización de los pasos formales, pero no por su identidad real.

En segundo lugar, en parte por el diseño, LP no suele saltarse los “grandes pasos” por su cuenta. Así que las partes interesantes de la prueba (por ej. su estructura) suelen terminar en el script de prueba. Si es afortunado, las minucias superfluas no inundarán las partes interesantes.

En tercer lugar, confiamos en que LP lidiará perfectamente el trabajo más fuerte. Es posible que LP no sea muy “brillante”. Es probable que no llegue al final de una prueba por sí sólo. No obstante, nos fiamos de la firmeza de los pasos que toma.

En cierto modo, este tercer punto distingue el área completa de investigación relativa a los demostradores de teoremas y asistentes automatizados de demostración de teoremas, del tipo de prueba asistida por ordenador realizada por Appel and Haken. En la prueba del teorema de los cuatro colores, Appel y Haken utilizaban un programa computacional especializado (y según mi antiguo tutor, David Gries, pobremente estructurado) para verificar ciertos casos de la prueba. El simple hecho de que este programa estuviese escrito como parte de la prueba, hace que la verificación del programa sea parte de la verificación de la prueba.

Por el contrario, los demostradores de teoremas se utilizan, se examinan, se extienden y se depuran repetidas veces durante años. El simple hecho de que exista una comunidad de usuarios para una herramienta en concreto, es un seguro de que ésta realiza sus tareas asignadas tal y como se espera. De algún modo, la corrección del mismo demostrador de teoremas es una conjetura separable de los teoremas con la que ésta se ha utilizado para revisar. ¿Constituye en realidad este conjunto de pruebas para la corrección de un demostrador de teoremas, una prueba en el sentido matemático? Probablemente no. Sin embargo, por esta misma razón, tampoco la revisión por iguales y publicación de una prueba “afirmada” constituye una prueba definitiva de que la publicación es, de hecho, correcta.

De algún modo, los scripts de pruebas LP son más sencillos de revisar que los enormes tratados matemáticos. Tengo dudas, por ejemplo, acerca de si sería más fácil para mi componer mi propio asistente demostrador de teoremas para volver a revisar un script, que tener que verificar los pasos en la prueba de Andrew Wiles sobre la conjetura de Taniyama-Shimura.

¿Cumplen los scripts de pruebas LP funciones explicativas? Es posible. Durante un tiempo, nuestro grupo de investigación ha estado desarrollando un conjunto de tecnología de pruebas en un sentido matemático. Como resultado, las pruebas del sistema distribuido que nuestro grupo produce siguen un formato muy estilizado. (Este formato estilizado es lo que ha empujado a la automatización de las técnicas de prueba). Como resultado, la parte de la prueba termina entonces en el script de prueba LP, que suele ser la parte que distingue a esta prueba en concreto de las anteriores. Por tanto, para el ojo educado, el script de la prueba subraya realmente las partes interesantes de la prueba.

No obstante, admitiría que no todas las pruebas tienen que ser explicativas.

Algunas son, de hecho, fines prácticos en sí. Esto es, demostrar que un sistema es correcto puede ser sólo interesante en la medida en que uno se preocupa de que éste sea correcto y de que no tenga mayores implicaciones para la construcción de más estructuras matemáticas.

En estos casos, los scripts de prueba se convierten en algo muy práctico. Concretamente, cuando el diseño de un sistema cambia, resulta a menudo muy fácil volver a verificar el sistema utilizando LP, cuando incluso adaptar una prueba informal a mano puede ser sumamente engorroso.