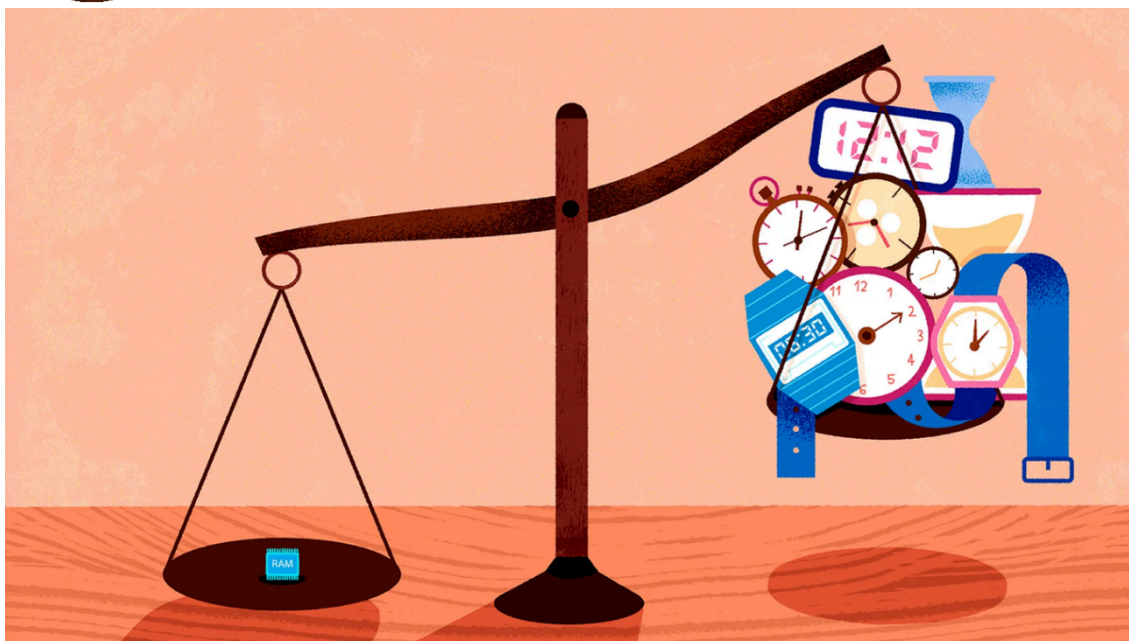


Para los algoritmos, un poco de memoria vale más que mucho tiempo



Por [Ben Brubaker](#)

Traducido por Víctor Hugo Casco



La “sorprendente” prueba de un científico informático, es el primer avance en 50 años en una de las preguntas más famosas de la informática.

Una tarde de julio de 2024, [Ryan Williams](#) se propuso demostrar que estaba equivocado. Habían pasado dos meses desde que había descubierto algo sorprendente sobre la relación entre el tiempo y la memoria en informática. Era un esbozo de una prueba matemática de que la memoria era más poderosa de lo que creían los informáticos: una pequeña cantidad sería tan útil como mucho tiempo en todos los cálculos imaginables. Eso sonaba tan improbable que asumió que algo andaba mal, y rápidamente dejó la prueba a un lado para centrarse en ideas menos disparatadas. Ahora, por fin, había encontrado tiempo para encontrar el error.

Eso no fue lo que pasó. Tras horas de analizar su argumento, Williams no pudo encontrarle ni un solo fallo.

“Pensé que me estaba volviendo loco”, comenta Williams, informático teórico del Instituto Tecnológico de Massachusetts (MIT). Por primera vez, empezó a considerar la posibilidad de que tal vez, solo tal vez, la memoria fuera tan poderosa como sugería su trabajo.

Durante los meses siguientes, perfeccionó los detalles, analizó cada paso con detalle y solicitó la opinión de otros investigadores. En febrero, finalmente [publicó su prueba en línea](#), con gran éxito.

"Es increíble. Es hermoso", dijo [Avi Wigderson](#), informático teórico del Instituto de Estudios Avanzados de Princeton, Nueva Jersey. En cuanto se enteró de la noticia, Wigderson le envió a Williams un correo electrónico de felicitación. El asunto decía: "Me dejaste alucinado".

El tiempo y la memoria (también llamada espacio) son los dos recursos más fundamentales en computación: todo algoritmo tarda un tiempo en ejecutarse y requiere espacio para almacenar datos mientras se ejecuta. Hasta ahora, los únicos algoritmos conocidos para realizar ciertas tareas requerían una cantidad de espacio aproximadamente proporcional a su tiempo de ejecución, y los investigadores habían asumido durante mucho tiempo que no había forma de mejorarlo. La demostración de Williams estableció un procedimiento matemático para transformar cualquier algoritmo, independientemente de su función, en una forma que utilice mucho menos espacio.



Ryan Williams sorprendió a sus colegas con una prueba histórica sobre la relación entre el tiempo y el espacio en la computación. Katherine Taylor para la revista *Quanta*

Es más, este resultado —una afirmación sobre lo que se puede calcular dada cierta cantidad de espacio— también implica un segundo resultado, sobre lo que no se puede calcular en un tiempo determinado. Este segundo resultado no es sorprendente en sí mismo: los investigadores esperaban que fuera cierto, pero no

tenían ni idea de cómo demostrarlo. La solución de Williams, basada en su primer resultado contundente, resulta casi caricaturescamente excesiva, similar a probar la culpabilidad de un presunto asesino estableciendo una coartada irrefutable para todos los demás habitantes del planeta. También podría ofrecer una nueva forma de abordar uno de los problemas abiertos más antiguos de la informática.

“Es un resultado bastante sorprendente y un avance enorme”, dijo [Paul Beame](#), informático de la Universidad de Washington. *“Es un poco menos sorprendente que sea Ryan quien lo hace”*.

Espacio para vagar

Williams, de 46 años, tiene un rostro abierto y expresivo, y un ligero toque de cana en el pelo. Su oficina, con vistas a las coloridas torres del Stata Center del MIT, es otro ejemplo del uso creativo del espacio. Un par de esterillas de yoga han transformado el alféizar de una ventana en un rincón de lectura improvisado, y el escritorio está escondido en una esquina de forma peculiar, dejando espacio para un sofá frente a una gran pizarra blanca llena de garabatos matemáticos.

El MIT está muy lejos del hogar de Williams en la Alabama rural, donde abundaba el espacio. Creció en una granja de 20 hectáreas y vio por primera vez una computadora a los 7 años, cuando su madre lo llevó en coche por todo el condado para una clase especial de refuerzo académico. Recordó haber quedado fascinado con un sencillo programa para generar un espectáculo digital de fuegos artificiales.

“Tomaba un color aleatorio y lo enviaba en una dirección aleatoria desde el centro del monitor”, dijo Williams. *“No se podía predecir la imagen que se obtendría”*. El mundo de las computadoras parecía un campo de juego salvaje y maravilloso, lleno de infinitas posibilidades. El joven Williams estaba fascinado.

“Escribía programas para mí mismo, en papel, para ejecutarlos en una computadora del futuro”, dijo. *“Mis padres no sabían qué hacer conmigo”*.

A medida que crecía, Williams pasó de las computadoras imaginarias a las reales. Durante sus dos últimos años de secundaria, se trasladó a la Escuela de Matemáticas y Ciencias de Alabama, un prestigioso internado público, donde tuvo su primer contacto con la teoría de la informática.

“Me di cuenta de que existía un mundo más amplio y de que existía una manera de pensar matemáticamente en las computadoras”, dijo. *“Eso era lo que quería hacer”*.

Williams se sintió especialmente atraído por una rama de la informática teórica llamada teoría de la complejidad computacional. Esta se ocupa de los recursos (como el tiempo y el espacio) necesarios para resolver problemas computacionales como ordenar listas o factorizar números. La mayoría de los problemas pueden resolverse mediante diversos algoritmos, cada uno con sus propias exigencias de tiempo y espacio. Los teóricos de la complejidad clasifican los problemas en

categorías, llamadas clases de complejidad, según las demandas de recursos de los mejores algoritmos para resolverlos; es decir, los algoritmos que se ejecutan más rápido o que utilizan menos espacio.

Pero ¿cómo lograr que el estudio de los recursos computacionales sea matemáticamente riguroso? No se llegará muy lejos si se intenta analizar el tiempo y el espacio comparando minutos con megabytes. Para avanzar, es necesario partir de las definiciones correctas.

Ser ingenioso

Estas definiciones surgieron del trabajo de Juris Hartmanis, un informático pionero con experiencia en la gestión de recursos limitados. Nació en 1928 en el seno de una prominente familia letona, pero su infancia se vio interrumpida por el estallido de la Segunda Guerra Mundial. Las fuerzas soviéticas de ocupación arrestaron y ejecutaron a su padre, y después de la guerra, Hartmanis terminó la secundaria en un campo de refugiados. Continuó sus estudios universitarios, donde destacó a pesar de no poder permitirse libros de texto.

“Lo compensaba tomando apuntes muy detallados en las clases”, recordó en una [entrevista de 2009](#). *“Hay cierta ventaja en improvisar y superar las dificultades”*. Hartmanis emigró a Estados Unidos en 1949 y realizó diversos trabajos —construyendo maquinaria agrícola, fabricando acero e incluso sirviendo de mayordomo— mientras estudiaba matemáticas en la Universidad de Kansas City. Continuó su carrera profesional con un éxito espectacular en el joven campo de la informática teórica.

En 1960, mientras trabajaba en el laboratorio de investigación de General Electric en Schenectady, Nueva York, Hartmanis conoció a Richard Stearns, un estudiante de posgrado que realizaba prácticas de verano. En un par de [artículos pioneros](#), establecieron definiciones matemáticas precisas de tiempo y espacio. Estas definiciones proporcionaron a los investigadores el lenguaje necesario para comparar ambos recursos y clasificar los problemas en clases de complejidad.

Una de las clases más importantes se conoce con el nombre modesto de "P". En términos generales, abarca todos los problemas que pueden resolverse en un tiempo razonable. Una clase de complejidad análoga para el espacio se denomina "PSPACE".

La relación entre estas dos clases es una de las cuestiones centrales de la teoría de la complejidad. Todo problema en P también está en PSPACE, porque los algoritmos rápidos simplemente no tienen tiempo suficiente para ocupar mucho espacio en la memoria de una computadora. Si la afirmación inversa también fuera cierta, las dos clases serían equivalentes: el espacio y el tiempo tendrían una potencia computacional comparable. Pero los teóricos de la complejidad sospechan que

PSPACE es una clase mucho más grande, que contiene muchos problemas que no están en P. En otras palabras, creen que el espacio es un recurso computacional mucho más poderoso que el tiempo. Esta creencia se deriva del hecho de que los algoritmos pueden usar la misma pequeña porción de memoria una y otra vez, mientras que el tiempo no es tan indulgente: una vez que pasa, no se puede recuperar.

“La intuición es muy simple”, dijo Williams. “Se puede reutilizar el espacio, pero no el tiempo”.

Pero la intuición no basta para los teóricos de la complejidad: buscan pruebas rigurosas. Para demostrar que PSPACE es mayor que P, los investigadores tendrían que demostrar que, para algunos problemas en PSPACE, los algoritmos rápidos son categóricamente imposibles. ¿Por dónde empezarían?

Una odisea del espacio

Dio la casualidad de que comenzaron en la Universidad de Cornell, adonde Hartmanis se trasladó en 1965 para dirigir el recién creado departamento de informática. Bajo su liderazgo, este se convirtió rápidamente en un centro de investigación en teoría de la complejidad, y a principios de la década de 1970, dos investigadores de la Universidad, John Hopcroft y Wolfgang Paul, se propusieron establecer un vínculo preciso entre el tiempo y el espacio.

Hopcroft y Paul sabían que para resolver el problema de P versus PSPACE, tendrían que demostrar que no se pueden realizar ciertos cálculos en un tiempo limitado. Pero es difícil demostrar una negativa. En cambio, decidieron invertir el problema y explorar qué se puede hacer con un espacio limitado. Esperaban demostrar que los algoritmos, con un presupuesto de espacio determinado, pueden resolver los mismos problemas que los algoritmos con un presupuesto de tiempo ligeramente mayor. Esto indicaría que el espacio es al menos ligeramente más poderoso que el tiempo: un paso pequeño pero necesario para demostrar que PSPACE es mayor que P.

Con ese objetivo en mente, recurrieron a un método que los teóricos de la complejidad llaman simulación, que consiste en transformar algoritmos existentes en nuevos que resuelven los mismos problemas, pero con diferentes cantidades de espacio y tiempo. Para comprender la idea básica, imagina que te dan un algoritmo rápido para ordenar alfabéticamente tu estantería, pero que requiere que coloques tus libros en docenas de pilas pequeñas. Quizás prefieras un enfoque que ocupe menos espacio en tu apartamento, aunque lleve más tiempo. Una simulación es un procedimiento matemático que podrías usar para obtener un algoritmo más adecuado: introdúcelo en el original y obtendrás un nuevo algoritmo que ahorra espacio a costa de tiempo.

Los diseñadores de algoritmos han estudiado durante mucho tiempo estas compensaciones espacio-temporales para tareas específicas como la ordenación. Pero para establecer una relación general entre tiempo y espacio, Hopcroft y Paul necesitaban algo más completo: un procedimiento de simulación que ahorrara espacio y fuera compatible con cualquier algoritmo, independientemente del problema que resolviera. Esperaban que esta generalidad tuviera un coste. Una simulación universal no puede explotar los detalles de ningún problema específico, por lo que probablemente no ahorraría tanto espacio como una simulación especializada. Pero cuando Hopcroft y Paul comenzaron su trabajo, no se conocían métodos universales para ahorrar espacio. Incluso un pequeño ahorro sería un avance.

El gran avance se produjo en 1975, después de que Hopcroft y Paul se asociaran con un joven investigador llamado [Leslie Valiant](#). El trío ideó un [procedimiento de simulación universal](#) que siempre ahorra algo de espacio. Independientemente del algoritmo que se le asigne, se obtendrá uno equivalente cuyo presupuesto de espacio es ligeramente menor que el presupuesto de tiempo del algoritmo original.

«Todo lo que se puede hacer en tan poco tiempo, también se puede hacer en un espacio ligeramente menor», dijo Valiant. Fue el primer gran paso en la búsqueda de conectar el espacio y el tiempo.

Pero entonces el progreso se estancó, y los teóricos de la complejidad comenzaron a sospechar que se habían topado con una barrera fundamental. El problema residía precisamente en el carácter universal de la simulación de Hopcroft, Paul y Valiant. Si bien muchos problemas pueden resolverse con mucho menos espacio que tiempo, algunos parecían necesitar casi tanto espacio como tiempo. De ser así, las simulaciones universales más eficientes en cuanto al espacio serían imposibles. Paul y otros dos investigadores pronto demostraron que, [en efecto, son imposibles](#), siempre que se parta de una suposición aparentemente obvia: diferentes fragmentos de datos no pueden ocupar el mismo espacio en la memoria al mismo tiempo.

“Todo el mundo daba por sentado que no se podía hacer nada mejor”, dijo Wigderson.

El resultado de Paul sugería que resolver el problema de P versus PSPACE requeriría abandonar por completo la simulación en favor de un enfoque diferente, pero nadie tenía buenas ideas. Así se mantuvo el problema durante 50 años, hasta que Williams finalmente lo despejó.

Pero primero tuvo que terminar la universidad.

Clases de complejidad

En 1996, llegó el momento de que Williams solicitara ingreso a la universidad. Sabía que estudiar teoría de la complejidad lo llevaría lejos de casa, pero sus padres le dejaron claro que la Costa Oeste y Canadá estaban descartadas. Entre las opciones que le quedaban, Cornell destacaba por su lugar destacado en la historia de su disciplina favorita.

«Este tipo, Hartmanis, definió las clases de complejidad temporal y espacial», recordó haber pensado. *«Eso fue importante para mí».*

Williams fue admitido en Cornell gracias a una generosa ayuda financiera y llegó en el otoño de 1997, con la intención de hacer lo que fuera necesario para convertirse en teórico de la complejidad. Su tenacidad impresionó a sus compañeros.

“Él estaba totalmente metido en la teoría de la complejidad”, cuenta [Scott Aaronson](#), un científico informático de la Universidad de Texas en Austin, que coincidió con Williams en Cornell.

Pero para el segundo año, Williams tenía dificultades para mantenerse al día en cursos que priorizaban el rigor matemático sobre la intuición. Tras obtener una calificación regular en una clase de teoría de la computación, el profesor le sugirió que considerara otras carreras. Pero Williams no renunciaría a su sueño. Insistió y se matriculó en un curso de teoría de posgrado, con la esperanza de que una excelente calificación en la clase más difícil le valiera una buena calificación en su solicitud de posgrado. El profesor que impartía ese curso de posgrado era Hartmanis, para entonces un veterano experto en el campo.

Williams comenzó a asistir a las horas de consulta de Hartmanis todas las semanas, donde casi siempre era el único estudiante presente. Su tenacidad dio sus frutos: obtuvo una calificación de sobresaliente en el curso, y Hartmanis aceptó asesorarlo en un proyecto de investigación independiente el semestre siguiente. Sus reuniones semanales continuaron durante toda la etapa universitaria de Williams. Hartmanis lo animó a cultivar un enfoque individualizado en la investigación de la complejidad y lo alejó con delicadeza de los callejones sin salida.

“Me influyó profundamente entonces”, dijo Williams. *“Supongo que todavía lo siento ahora”.*

Pero a pesar de obtener una codiciada beca de investigación de posgrado de la Fundación Nacional de Ciencias, Williams fue rechazado en todos los programas de doctorado a los que solicitó admisión. Al enterarse de la noticia, Hartmanis llamó a un colega y luego felicitó a Williams por haber sido aceptado en un programa de maestría de un año en Cornell. Un año después, Williams volvió a solicitar admisión a varios programas de doctorado, y con esa experiencia investigadora adicional, alcanzó el éxito.

Williams continuó trabajando en teoría de la complejidad durante sus estudios de posgrado y en los años posteriores. En 2010, cuatro años después de obtener su doctorado, logró un [resultado trascendental](#): un pequeño paso, pero el más grande en décadas, hacia la solución de la [pregunta más famosa de la informática teórica](#): la naturaleza de los problemas complejos. Este resultado consolidó la reputación de Williams, quien posteriormente escribió docenas de artículos sobre diferentes temas de teoría de la complejidad.

P versus PSPACE no era una de ellas. Williams había estado fascinado por el problema desde que lo abordó en la universidad. Incluso complementó su currículo de informática con cursos de lógica y filosofía, buscando inspiración en otras perspectivas del tiempo y el espacio, sin éxito.

“Siempre lo he tenido en mente”, comenta Williams. “No se me ocurría nada lo suficientemente interesante como para decirlo”.

El año pasado, finalmente encontró la oportunidad que había estado esperando.

Guijarros blandos

La historia del nuevo resultado de Williams comenzó con el progreso en una pregunta diferente sobre la memoria en computación: ¿Qué problemas se pueden resolver con un espacio extremadamente limitado? En 2010, el pionero teórico de la complejidad Stephen Cook y sus colaboradores inventaron una tarea, llamada el [problema de evaluación del árbol](#), que demostraron que sería imposible para cualquier algoritmo con un presupuesto de espacio inferior a un umbral específico. Pero existía una laguna. La demostración se basaba en la misma suposición de sentido común que Paul y sus colegas habían hecho décadas antes: los algoritmos no pueden almacenar datos nuevos en un espacio ya lleno.

Durante más de una década, los teóricos de la complejidad intentaron resolver esa laguna. En 2023, [el hijo de Cook, James](#), y el investigador [Ian Mertz](#) la desvelaron por completo al idear [un algoritmo](#) que resolvía el problema de la evaluación de árboles utilizando [mucho menos espacio](#) del que se creía posible. La demostración de Cook padre suponía que los bits de datos eran como piedritas que deben ocupar lugares separados en la memoria de un algoritmo. Pero resulta que esa no es la única forma de almacenar datos. *«De hecho, podemos pensar en estas piedritas como cosas que podemos aplastar ligeramente unas sobre otras»*, dijo Beame.

El algoritmo de Cook y Mertz despertó la curiosidad de muchos investigadores, pero no estaba claro que tuviera aplicaciones más allá del problema de la evaluación de árboles. *«Nadie vio su importancia para la comparación entre tiempo y espacio»*, afirmó Wigderson.

Ryan Williams fue la excepción. En la primavera de 2024, un grupo de estudiantes presentó el artículo de Cook y Mertz como proyecto final en una clase que él

impartía. Su entusiasmo lo inspiró a analizarlo con más detalle. En cuanto lo hizo, se le ocurrió una idea. Se dio cuenta de que el método de Cook y Mertz era en realidad una herramienta de uso general para reducir el uso del espacio. ¿Por qué no usarlo para impulsar una nueva simulación universal que vinculara el tiempo y el espacio, como la diseñada por Hopcroft, Paul y Valiant, pero mejorada?

Ese resultado clásico era una forma de transformar cualquier algoritmo con un presupuesto de tiempo determinado en un nuevo algoritmo con un presupuesto de espacio ligeramente menor. Williams observó que una simulación basada en guijarros blandos reduciría considerablemente el uso de espacio del nuevo algoritmo, aproximadamente igual a la raíz cuadrada del presupuesto de tiempo del algoritmo original. Ese nuevo algoritmo, eficiente en cuanto a espacio, también sería mucho más lento, por lo que era improbable que la simulación tuviera aplicaciones prácticas. Pero desde un punto de vista teórico, fue nada menos que revolucionario.

Durante 50 años, los investigadores asumieron que era imposible mejorar la simulación universal de Hopcroft, Paul y Valiant. La idea de Williams, si funcionaba, no solo batiría su récord, sino que lo demolería.

“Lo pensé y pensé: 'Bueno, eso simplemente no puede ser cierto'”, dijo Williams. Lo dejó a un lado y no volvió a retomarlo hasta ese fatídico día de julio, cuando intentó encontrar la falla en el argumento y fracasó. Tras darse cuenta de que no había falla alguna, pasó meses escribiendo y reescribiendo la prueba para que fuera lo más clara posible.

A finales de febrero, Williams finalmente [publicó el artículo terminado en línea](#). Cook y Mertz estaban tan sorprendidos como todos los demás. *«Tuve que dar un largo paseo antes de hacer nada más»,* dijo Mertz.

Valiant tuvo un adelanto de la mejora de Williams con respecto a su resultado de hace décadas durante su viaje matutino al trabajo. Durante años, ha impartido clases en la Universidad de Harvard, muy cerca de la oficina de Williams en el MIT. Se conocían de antes, pero no sabían que vivían en el mismo barrio hasta que se encontraron en el autobús un nevado día de febrero, unas semanas antes de que se hiciera público el resultado. Williams le describió su prueba al sorprendido Valiant y le prometió enviarle su trabajo.

“Me impresionó muchísimo”, dijo Valiant. *“Si obtienes el mejor resultado matemático en 50 años, es que algo estás haciendo bien”.*

PSPACE: La última frontera

Con su nueva simulación, Williams demostró un resultado positivo sobre el poder computacional del espacio: los algoritmos que utilizan relativamente poco espacio pueden resolver todos los problemas que requieren un tiempo ligeramente mayor.

Luego, con solo unas pocas líneas de cálculo, invirtió esta teoría y demostró un resultado negativo sobre el poder computacional del tiempo: al menos algunos problemas no pueden resolverse a menos que se utilice más tiempo que espacio. Este segundo resultado, más limitado, coincide con lo que esperaban los investigadores. Lo curioso es cómo Williams llegó a esta conclusión: primero demostró un resultado aplicable a todos los algoritmos, independientemente de los problemas que resuelvan.

“Todavía me cuesta creerlo”, dijo Williams. “Parece demasiado bueno para ser verdad”.

Expresado en términos cualitativos, el segundo resultado de Williams podría parecer la solución largamente buscada al problema de P versus PSPACE. La diferencia radica en la escala. P y PSPACE son clases de complejidad muy amplias, mientras que los resultados de Williams operan a un nivel más preciso. Estableció una brecha cuantitativa entre la potencia del espacio y la potencia del tiempo, y para demostrar que PSPACE es mayor que P, los investigadores tendrán que ampliar esa brecha muchísimo más.

Es un reto abrumador, similar a abrir una grieta en la acera con una palanca hasta que tenga el ancho del Gran Cañón. Pero podría lograrse usando una versión modificada del procedimiento de simulación de Williams que repite el paso clave muchas veces, ahorrando espacio cada vez. Es como aumentar la longitud de la palanca repetidamente: si la haces lo suficientemente grande, puedes abrir cualquier cosa. Esta mejora repetida no funciona con la versión actual del algoritmo, pero los investigadores desconocen si se trata de una limitación fundamental.

“Podría ser un cuello de botella definitivo, o podría ser un cuello de botella de 50 años”, dijo Valiant. “O podría ser algo que tal vez alguien pueda resolver la próxima semana”.

Si el problema se resuelve la semana que viene, Williams se lamentará. Antes de escribir el artículo, pasó meses intentando, sin éxito, ampliar su resultado. Pero incluso si dicha ampliación no es posible, Williams confía en que una mayor exploración espacial sin duda conducirá a algo interesante, quizás a avances en un problema completamente diferente.

«Nunca puedo demostrar con precisión lo que quiero demostrar», dijo. «Pero a menudo, lo que demuestro es mucho mejor de lo que quería».

Nota del editor: Scott Aaronson es miembro del [consejo asesor](#) de Quanta Magazine .