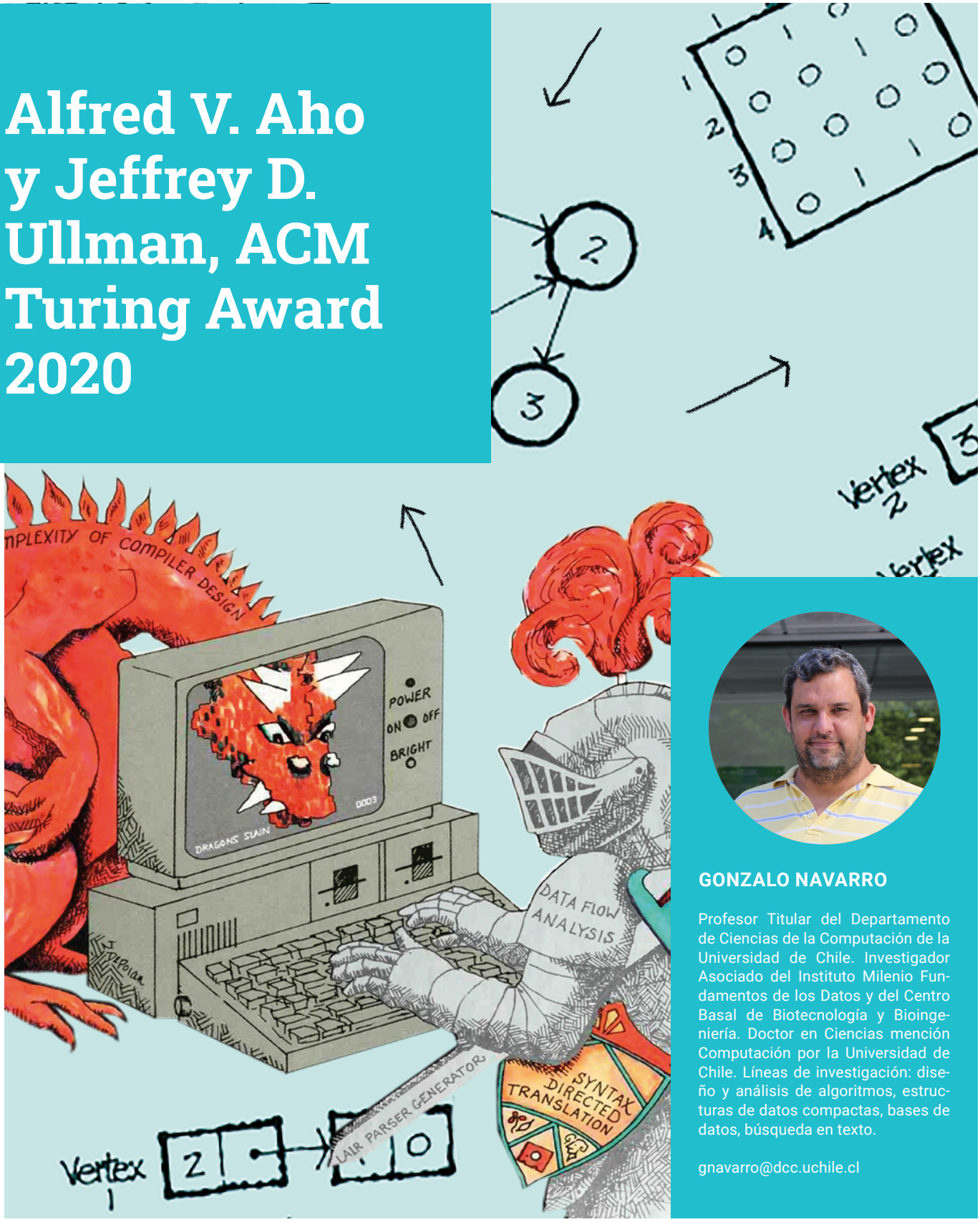
Alfred V. Aho y Jeffrey D. Ullman, ACM Turing Award 2020



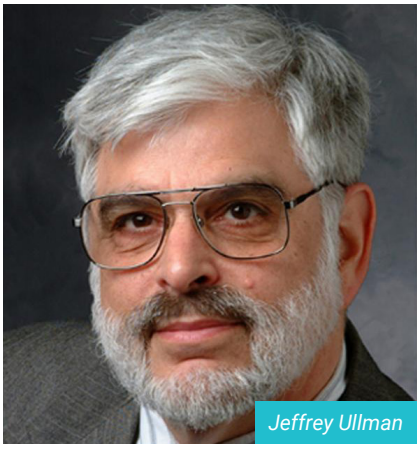
GONZALO NAVARRO

Profesor Titular del Departamento de Ciencias de la Computación de la Universidad de Chile. Investigador Asociado del Instituto Milenio Fundamentos de los Datos y del Centro Basal de Biotecnología y Bioingeniería. Doctor en Ciencias mención Computación por la Universidad de Chile. Líneas de investigación: diseño y análisis de algoritmos, estructuras de datos compactas, bases de datos, búsqueda en texto.

gnavarro@dcc.uchile.cl

RESUMEN. El que Aho y Ullman hayan sido distinguidos en 2020 con el ACM Turing Award, más que una sorpresa, se siente como un "¡ya era hora!". Sus nombres aparecen asociados a las ramas clave de la computación desde los años sesenta, en que la disciplina dejaba de ser una curiosidad y se empezaba a tomar en serio.

Aho y Ullman realizaron contribuciones fundamentales en la teoría, implementación y divulgación en el área de lenguajes formales, compilación, algoritmos, lenguajes de programación y bases de datos. En un estilo poco común hoy en día, muchas de sus contribuciones se encuentran en forma de libros donde recopilaron todo lo que había que saber, y que han sido claves en la formación de millones de programadores e investigadores por décadas. Parafraseando a Aho, sus técnicas llevaron la compilación de ser un arte a ser una ciencia. ¡Y no son personajes obsoletos! Hoy en día se interesan en temas como integración de datos, paralelismo masivo o computación cuántica.



Fuente: https://awards.acm.org/about/2020-turing

Ganadores del Premio Turing 2020.

Una de las experiencias más hermosas de las que tengo memoria al estudiar Computación en los ochenta es aquella perfecta combinación de lenguajes formales, *parsing*, compilación, semántica, optimización de código, lenguajes de programación y algoritmos. Estos temas formaban un todo armónico que siempre percibí como el núcleo científico de nuestra maravillosa disciplina, posiblemente con el agregado de bases de datos. Los nombres de Aho, Hopcroft y Ullman aparecían donde fuera que uno se asomara, como una versión siglo XX de aquellos últimos universalistas del Renacimiento que sabían todo de todo, y que eran los padres fundadores de todo el conocimiento. El que Aho y Ullman hayan sido distinguidos en 2020 con el ACM Turing Award (Hopcroft lo fue en 1986), más que una sorpresa, se siente como un "¡ya era hora!" para estos todavía activos fundadores de la Computación moderna.

No es casualidad que ambos recibieran juntos la distinción más alta de nuestra disciplina. El comienzo de su historia académica se remonta a la época en que la Computación dejaba de ser una curiosidad y se empezaba a tomar en serio. Ambos se habían graduado en 1963 (Aho en la Universidad de Toronto, Ullman en la Universidad de Columbia) y se conocieron el mismo año, en la fila de admisión del Doctorado en Ingeniería Eléctrica y Computación de la Universidad de Princeton. Fue el comienzo de una prolífica amistad y colaboración científica que duraría muchas décadas, y que literalmente le daría forma a nuestra disciplina como la conocemos hoy.

Corrieron suerte distinta en sus doctorados: aparentemente Ullman lo tuvo más fácil. Aho recuerda que "Jeff había encontrado su tema de tesis al comienzo de su segundo año, y se pasó todo el tercer año tipeando su tesis, página por página. Mientras tanto, yo me tiraba de los pelos porque todavía no tenía un tema". Para peor, el profesor guía que había reclutado a Aho se fue a trabajar a la Universidad de Stanford, dejándolo huérfano. Como una confirmación de que no hay mal que por bien no venga, el nuevo profesor guía que encontró Aho fue John Hopcroft.

Ullman se doctoró en 1966, con una tesis titulada "Synchronization Error Correcting Codes", y Aho un año después, con la tesis "Indexed Grammars, an Extension of Context Free Grammars". Ullman publicó sus primeros resultados en el *IBM Journal of Research and Development* (1965) y en las *IEEE Transactions on Information Theory* (1966), mientras que Aho publicó su tesis en *SWAT'67* (hoy *FOCS*), y la versión de revista en el *Journal of the ACM* en 1968.

Apenas doctorados, ambos se fueron a trabajar a los Bell Laboratories e inmediatamente comenzaron a colaborar en el desarrollo de algoritmos eficientes para el *parsing* y compilación de lenguajes de programación. No era una casualidad. Estamos en los "míticos años sesenta", en los que nacieron el sistema operativo Unix y el lenguaje de programación C, precisamente en los Bell Labs. Las publicaciones de sus resultados científicos también comienzan a aparecer desde 1968.

Asimismo, su vocación por resumir y divulgar el conocimiento muestra las primeras señales, con su bello survey "The Theory of Languages" en *Mathematical Systems Theory* (1968).

Ullman retornó pronto a Princeton, en 1969, y se mudó a la Universidad de Stanford en 1979, donde pasó a ser profesor emérito en 2002. Aho se quedó mucho más en Bell Labs, hasta que se mudó a la Universidad de Columbia en 1995, de donde también es profesor emérito. Su colaboración continuó ininterrumpidamente todo este tiempo, y es difícil exagerar el impacto que tuvo.

No es necesario explicar a alguien con formación en Computación la relevancia de los lenguajes de programación, que nos permiten expresar complejos procesos lógicos a un nivel suficientemente alto como para desarrollar el software tremendamente sofisticado que se requiere en un mundo cada vez más dependiente de la Computación. Los computadores no entienden estos lenguajes, sino unos mucho más elementales (el lenguaje de máquina), en los cuales la programación de tareas relativamente sencillas es un desafío intelectual formidable. Seríamos incapaces de hacer funcionar el mundo moderno programando en lenguaje de máquina. Simplemente no tenemos la capacidad intelectual necesaria. El desarrollo de sistemas cada vez más elaborados y complejos ha sido posible solo gracias al desarrollo de los lenguajes de programación llamados de alto nivel, y de otros lenguajes de aún más alto nivel por sobre ellos, como lenguajes para manipular bases de datos (SQL, SPARQL), para realizar procesamiento paralelo (MapReduce), para realizar procesamiento matemático y estadístico (Maple, R), y muchos otros.

Los sofisticados sistemas a los que estamos acostumbrados hoy en día utilizan varios niveles de lenguajes: un lenguaje gráfico para acceder a bases de datos se compila a SQL, el cual a su vez se traduce a C, y este finalmente a lenguaje de máquina. ¡Incluso escribimos artículos en LaTeX, un lenguaje que se compila a Tex, y este a su vez se traduce a PDF! Para que esta arquitectura funcione, es fundamental poder traducir eficientemente un programa escrito en un lenguaje de alto nivel a otro de más bajo nivel (ver Figura 1). Los años sesenta vieron surgir los primeros lenguajes de programación de alto nivel, y el problema de cómo traducirlos eficientemente a lenguaje de máquina era crucial para el desarrollo de la disciplina. Pero este era un problema complejo. Debe existir un conjunto de reglas sintácticas para escribir programas correctos, y esas reglas en sí se escriben en un lenguaje formal, por ejemplo con gramáticas libres del contexto. Debe diseñarse un mecanismo automático, un *parsing*, que utilice las reglas sintácticas y convierta el programa en una representación interna a partir de la cual se pueda generar un programa equivalente en lenguaje de máquina (ver Figura 2). Debe existir una semántica clara que describa qué hace exactamente un programa de alto nivel, de forma de asegurarse de traducirlo correctamente a lenguaje de máquina e incluso optimizarlo para que ejecute más rápido sin cambiar lo que debería hacer. Y todo esto debe hacerse eficientemente para poder traducir programas complejos de miles de líneas.

Aho y Ullman realizaron contribuciones fundamentales en la teoría, implementación y divulgación en el área de lenguajes formales, *parsing*, compilación, y algoritmos y su análisis, estableciendo además importantes conexiones entre ellas. En un estilo que es menos común hoy en día, muchas de sus contribuciones se encuentran en forma de

En los ochenta los nombres de Aho, Hopcroft y Ullman aparecían donde fuera que uno se asomara, como una versión siglo XX de aquellos últimos universalistas del Renacimiento que sabían todo de todo.

Figura 1. Proceso de compilación.

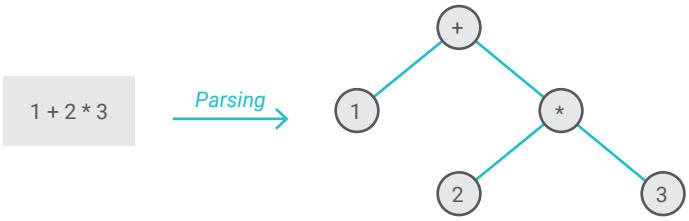


Figura 2. Ejemplo de *parsing* de una expresión aritmética.

Figura 3. Serie de libros de Aho y Ullman sobre compilación.

libros (ver Figuras 3 y 4) donde recopilaron lo que había que saber acerca de varias de estas áreas fundamentales, incluyendo técnicas desarrolladas por ellos y por otros. Estos libros han sido claves en la formación de millones de ingenieros de software e investigadores en Computación durante décadas, y en muchos casos son la forma más directa de exposición que hemos tenido todos nosotros a los logros de Aho y Ullman (y Hopcroft, en muchos casos). Parafraseando a Aho, estas técnicas llevaron la compilación de ser un arte a ser una ciencia. Aho también cuenta que la idea de escribir estos libros se la dio Richard Hamming, que también estaba en los Bell Labs. "Si quieren que su investigación tenga realmente impacto, deben enseñar a los demás a utilizarla, y la mejor forma de hacer eso es un libro", les dijo.

El más antiguo de estos libros es *"The Theory of Parsing, Translation, and Compiling"*: parte 1, Parsing, de 1972 y parte 2, Compiling, de 1973. Su sucesor, "*Principles of Compiler Design*" (1977), conocido como "el libro del dragón verde" por la ilustración de su tapa[1], contenía toda la teoría relevante a la compilación, desde lenguajes formales y parsing hasta diseño de compiladores, estableciendo un esquema claro y modular para llevar a cabo el proceso completo. Contiene también contribuciones algorítmicas de los autores para hacer el proceso más eficiente. Este libro y sus sucesivas ediciones se consideran la biblia del área y son usados en cursos universitarios y en la industria alrededor del mundo. La última edición de este libro, de 2007, se titula *"Compilers: Principles, Techniques and Tools"* e incluye como nuevos coautores a Monica Lam y Ravi Sethi.[2]

El siguiente libro más antiguo es el también famosísimo "*The Design and Analysis of Computer Algorithms*" de 1974, con Hopcroft, que fue por décadas el libro clásico en los cursos de Diseño y Análisis de Algoritmos en todas las universidades. El libro introduce, por ejemplo, el modelo RAM de computación (Random-Access Machine), que permite analizar los algoritmos de un modo realista para los computadores que utilizamos, y a la vez suficiente-

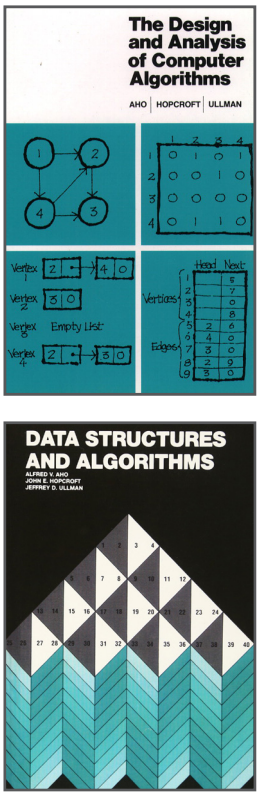


Figura 4. Libros de Aho y Ullman sobre algoritmos y estructuras de datos.

1 El dragón, una idea de Ullman, representaba la dificultad del problema, y era enfrentado por un caballero medieval. Cuenta Ullman que las tapas de los libros eran usualmente formales entonces, y que logró colar la idea. El presidente de Addison-Wesley vino luego a decirle que creía que era la peor tapa de la historia. A los tres meses, con las ventas disparadas, volvió para decirle que era la mejor.

2 La tapa de la edición de 1986 apareció fugazmente en la película "Hackers" en 1995. Aho cuenta que solo cuando vieron su libro ahí, sus hijos empezaron a pensar que su padre hacía algo interesante.

Además de sus nueve libros en conjunto, Aho y Ullman tienen una amplia producción de artículos científicos de gran impacto (Aho sobre 100, con 90 mil citas, y Ullman sobre 300, con 135 mil citas).

mente general para que el análisis sea factible y universal. El libro también contiene varias contribuciones propias de los autores. A pesar de que en décadas recientes han aparecido libros que son hoy más populares (como el de Cormen, Leiserson, Rivest y Stein), ¡algunas partes del curso CC4102 Diseño y Análisis de Algoritmos de nuestro Departamento todavía se basan en aquel libro! Un buen complemento que apareció después es "*Data Structures and Algorithms*" (1983). En su época, los conocíamos como "el Ajo verde" y "el Ajo negro" por el color de sus tapas (ver Figura 4) .

Además de sus nueve libros en conjunto y de varios otros escritos por separado, Aho y Ullman tienen una amplia producción de artículos científicos de gran impacto (Aho sobre 100, con 90 mil citas, y Ullman sobre 300, con 135 mil citas), así como software de amplísima difusión y utilidad. Es un ejercicio muy interesante pasear por sus publicaciones para descubrir los temas de interés de estos gigantes de la disciplina. No hay una que no toque un tema relevante, no hay compromiso entre calidad y cantidad. Además de sus trabajos conjuntos en teoría de la computación, compiladores y algoritmos, incursionaron en áreas como sistemas operativos (donde Aho profundizó más) y bases de datos (donde Ullman profundizó mucho más).

Un ejemplo de contribución algorítmica es su artículo conjunto (con Hopcroft) para encontrar el ancestro común más bajo de dos nodos en un árbol, un problema que aparece todo el tiempo al desarrollar algoritmos para datos jerárquicos. Ellos propusieron el problema y las primeras soluciones (*STOC*'73 y *SIAM Journal on Computing* 1976), la óptima (por Dov Harel y Robert Tarjan) tuvo que esperar hasta 1984. Otro ejemplo algorítmico es la cota inferior cuadrática para calcular la subsecuencia común más larga entre dos secuencias (con Daniel Hirschberg, *SWAT*'74 y *Journal of the ACM* 1976), un problema crucial en el alineamiento de genomas, detección de plagio y traducción automática. Un tercero es su algoritmo para representar eficientemente la clausura transitiva de un grafo (con Michael Garey, *SIAM Journal on Computing* 1972). En el área de lenguajes para bases de datos, un importante artículo conjunto es su estudio del poder expresivo del álgebra relacional, mostrando que los puntos fijos (por ejemplo, la relevante clausura transitiva) no son expresables, y proponiendo extensiones que lo permitieran (*POPL*'79). Este es un tema sumamente actual en bases de datos de grafos, por ejemplo, donde lenguajes como Datalog y SPARQL permiten clausuras transitivas. Finalmente, Aho destaca en su charla de recepción del Turing Award un artículo en el que muestran cómo describir la sintaxis de lenguajes en forma de gramáticas ambiguas más reglas de desambiguación, de una forma que puede trasladarse directamente a *parsers* eficientes (*POPL*'73 y *Communications of the ACM* 1975, con Stephen Johnson).

Aho trabajó por separado, por ejemplo, en problemas relacionados con búsqueda en texto, tales como el famoso algoritmo de Aho-Corasick (*Communications of the ACM* 1975, con Margaret Corasick) para buscar un gran número de patrones en un texto en tiempo lineal (e implementado por él en el software *fgrep* de Unix). Creó también famosos programas básicos y universales para Unix (y, hoy, Linux), como el software *egrep* de búsqueda de expresiones regulares y el software *awk* de manipulación de texto (la "a" es por Aho) que usan todo el tiempo los usuarios de Linux. Estas herramientas se usaron, por ejemplo, en el conocido software *lex*, que provee la base de análisis sintáctico para desarrollar *parsers* de lenguajes. *Lex* se usa en particular en el igualmente conocido generador de *parsers yacc*, desarrollado sobre las técnicas creadas por Aho, Hopcroft y Ullman. La combinación *lex/yacc*, o sus derivados como *flex/bison*, está en la base de la mayoría de los compiladores modernos.[3] En el área de lenguajes formales, el artículo de su tesis (*Journal of the ACM* 1968), donde propone un tipo de gramática más potente que las libres del contexto para describir lenguajes de programación y que es todavía procesable eficientemente, es uno de los más citados. Aho declara que sus intereses actuales son lenguajes de programación, compiladores, algoritmos y computación cuántica.

Ullman contribuyó en forma decisiva en el área de bases de datos, al punto de que se lo considera uno de los fundadores de la disciplina. Su libro "*Principles of Database Systems*" (1980) es fundacional. Ullman se ha mantenido a la par de la evolución del área, con contribuciones en minería de datos, datos semiestructurados, OLAP, integración de datos, bases de datos deductivas y ciencia de datos. Un ejemplo de sus contribuciones es su técnica de "*magic sets*" para evaluar reglas lógicas en

3 Aho comenta que con estas herramientas sus estudiantes desarrollaban un compilador para algún lenguaje acotado en grupos de cinco durante tan solo un semestre. Me recordó, con nostalgia, que nosotros hicimos lo mismo en esos años, compilando un lenguaje tipo Pascal al Assembler del Intel 8088, optimización incluida, ¡no es una exageración de Aho!

bases de datos, que las reescribe de forma que se puedan evaluar *bottom-up* eficientemente en base a *joins*. Si bien es una heurística, esta técnica es la base para optimizar estos programas (publicado con varios coautores en *PODS'*86). Su artículo más citado (*SIGMOD'*97, con Sergey Brin, Rajeev Motwani y Shalom Tsur) presenta nuevos algoritmos para analizar información de grandes volúmenes de compras pasando pocas veces sobre los datos, de modo de detectar conjuntos de ítems que se compran juntos frecuentemente. Este es un problema muy popular de minería de datos. En términos de tecnología, Ullman fue profesor guía de Sergey Brin (cofundador de Google) y formó parte del directorio técnico de la empresa. Finalmente, su libro "*Introduction to Automata Theory, Languages and Computation*" (1972, con Hopcroft) es uno de los más conocidos y usados en el área. Ullman declara que sus intereses actuales son teoría de bases de datos, integración de datos, minería de datos y educación en línea.

Es un ejercicio muy interesante pasear por sus publicaciones para descubrir los temas de interés de estos gigantes de la disciplina. No hay una que no toque un tema relevante, no hay compromiso entre calidad y cantidad.

A los lectores más jóvenes, que tal vez sientan que estamos hablando de caballeros medievales, los invito a ver la charla de recepción del Turing Award, que usa el interesante concepto de abstracción computacional como hilo conductor. Van a descubrir a dos personajes atemporales, que pasan en un instante de hablar de los inicios de los lenguajes de programación y del álgebra relacional a los temas más candentes de integración de datos, paralelismo masivo o computación cuántica. La entrevista también nos deja entrever algo de las personas detrás de los nombres: mientras Aho está a sus anchas en la entrevista y la disfruta, Ullman sufre visiblemente con cada pregunta. Al final, después de repasar las contribuciones de estos dos gigantes, es toda una lección de humildad el escuchar decir a Aho que simplemente fue afortunado por entrar a la disciplina cuando todavía era fácil hacer contribuciones fundamentales, o decir a Ullman (en otra entrevista) que siente un poco el "síndrome del impostor". ¡Qué nos queda a los demás! ■

Agradecimientos

Agradezco las sugerencias varias de Pablo Barceló y Éric Tanter.

REFERENCIAS

- Artículo de ACM sobre el Turing Award 2020.
  https://awards.acm.org/about/2020-turing
- Video en YouTube de la charla y entrevista asociadas al otorgamiento del premio.
  https://www.youtube.com/watch?v=ixIlknu7svM&ab_channel=AssociationforComputingMachinery%28ACM%29
- Artículo de la Universidad de Princeton sobre el Turing Award 2020.
  https://www.dailyprincetonian.com/article/2021/04/princeton-alumni-turing-award-aho-ullman-computer-science
- Páginas de DBLP de Aho y Ullman.
  https://dblp.org/pid/a/AVAho.html
  https://dblp.org/pid/u/JeffreyDUllman.html
- Páginas de Google Scholar de Aho y Ullman.
  https://scholar.google.cl/citations?user=gb2r2ssAAAAJ&hl=en&oi=ao
  https://scholar.google.cl/citations?user=wUJ2bXgAAAAJ&hl=en&oi=ao
- Páginas de Wikipedia de Aho y Ullman.
  https://en.wikipedia.org/wiki/Alfred_Aho
  https://en.wikipedia.org/wiki/Jeffrey_Ullman
- Páginas personales de Aho y Ullman.
  http://www.cs.columbia.edu/~aho/
  http://infolab.stanford.edu/~ullman/

Nota del editor:

Sobre las acusaciones a Ullman por su sostenido comportamiento discriminatorio

La siguiente columna refleja pura y exclusivamente la opinión y punto de vista del editor general de Revista Bits de Ciencia, y es independiente del artículo anterior.

Cada año, el anuncio del Premio Turing —el Premio Nobel de la computación— es esperado con mucha expectativa y entusiasmo por parte de toda la comunidad científica. En ese sentido, el anuncio del último año fue una rara excepción que sembró un manto de dudas sobre la pertinencia del proceso de selección. Nadie puso en duda el valor de las contribuciones científicas de los premiados, Aho y Ullman; sin embargo muchos levantaron una bandera roja por el sostenido comportamiento discriminatorio que Ullman ha mostrado hacia ciertos grupos a lo largo de las últimas décadas.

Los primeros signos de este tipo de comportamiento se remontan a mediados de los 2000 cuando, cansado de recibir y tener que responder correos de estudiantes iraníes, Ullman publicó su infame página web "Answers to All Questions Iranian", que mantuvo en línea hasta finales de 2020.[1] En la página web, entre otros, manifestaba abiertamente sus sentimientos contra el pueblo iraní y justificaba el genocidio sufrido por los pueblos nativos de Estados Unidos argumentando que "*esa es la manera en la que ocurren y siempre han ocurrido las cosas. Civilizaciones tecnológicamente más avanzadas reemplazan a civilizaciones menos avanzadas*".

Por otro lado, consideraba que los estudiantes que habían crecido en Irán luego de la Revolución Islámica de 1979 representaban un peligro para los Estados Unidos y no se les debía permitir estudiar en el país. Esto se ve reflejado en una discusión que mantuvo por Google+, apoyando el veto a los iraníes que la Universidad de Massachusetts estaba por instaurar en 2015, donde sostenía que *"tenemos que distinguir entre los estadounidenses de ascendencia iraní que han elegido hacer su camino junto a los Estados Unidos, y los iraníes que no se marcharon de Irán cuando los fanáticos religiosos se hicieron del control, y que pueden estar alineados con los deseos de Irán de construir armas nucleares y desplegar terroristas por todo el mundo. Si bien estoy convencido de que muchos de los estudiantes que viven en Irán no tienen mayor anhelo que marcharse del país mientras este siga gobernado por fundamentalistas islámicos, ¿podemos correr el riesgo de educarlos y que luego utilicen esa educación contra nosotros?"*.[2]

Un último ejemplo de la discriminación deliberada de Ullman hacia el pueblo iraní, en este caso bajo la presunción de ciertas posturas políticas, puede verse en una comunicación por correo electrónico que en 2010 mantuvo con un estudiante iraní sobre el proceso de admisión a la Universidad de Standford, de la cual Ullman era profesor emérito. A la consulta del estudiante, Ullman respondió: *"Incluso si estuviera en condiciones de ayudar, tampoco ayudaría a ningún estudiante iraní hasta que Irán no reconozca a Israel como tierra del pueblo judío. Sé que puede que no compartas la misma postura demencial de los mulás que gobiernan tu país, pero es una cuestión de principios. Si los iraníes quieren gozar de los beneficios de Stanford u otras instituciones de los Estados Unidos, deben respetar los valores que tenemos en los Estados Unidos, incluída la libertad de religión y el respeto a los derechos humanos"*.[3]

En respuesta a esto, el National Iranian American Council elevó una queja formal a la Universidad de Standford.[4] Sin embargo, alegando que las observaciones hechas por un académico ya retirado no reflejaban la postura de la Universidad ni su política de admisión, las autoridades de la Universidad decidieron no tomar ninguna medida al respecto, incluso permitiendo que Ullman siga manteniendo su página como parte del sitio web institucional.[5]

Al florecer todos estos antecedentes luego de la premiación, diversos miembros de la comunidad científica decidieron retirar el apoyo que inicialmente habían dado a Ullman. Entre ellos se destacan, por ejemplo, Shafi Goldwasser y Yann LeCun, antiguos ganadores del Premio Turing.[6] Este repudio se volvió mucho más masivo luego de la publicación de una carta abierta firmada por más de

1 https://web.archive.org/web/20200129080549/http:/infolab.stanford.edu/~ullman/pub/iranian.html.

2 https://www.ics.uci.edu/~eppstein/gplus/20150212-VDYSkY69tGe.html.

3 https://drive.google.com/file/d/1ZjydBnYeO8LhU3TyxvYxBTziWsQvSf2t/view.

4 https://web.archive.org/web/20110124051512/http:/www.niacouncil.org/site/DocServer/Stanford_Discrimination_Letter.pdf.

5 https://web.archive.org/web/20140809043733/http:/www.paaia.org/CMS/stanford-university-president-responds-directly-to-paaia-over-retired-professors-anti-iranian-remarks.aspx.

6 https://www.facebook.com/SimonsInstitute/posts/4003638433027737.

mil adherentes de la academia e industria, bajo la etiqueta *CSForInclusion*.[7]

Paradójicamente, la institución encargada de otorgar la premiación —la Association for Computing Machinery (ACM)—, se define como una *"organización científica y educativa global dedicada a promocionar el arte, ingeniería, ciencia y aplicaciones de la computación [...] promoviendo los más altos estándares éticos y profesionales"*, además de declarar a la diversidad, igualdad e inclusión como uno de sus cuatro valores fundamentales.[8]

Ante el rechazo público por la controvertida premiación, la ACM primero reaccionó con un escueto tweet diciendo que *"las afirmaciones hechas en el pasado por Jeffrey Ullman [...] no reflejan la visión de la ACM"*.[9] Luego publicó en su sitio web una respuesta oficial a la masiva carta en la que explicaba que la ACM nunca había recibido ninguna denuncia contra Ullman, que los miembros del comité encargado de la premiación se enteraron de sus antecedentes por la polémica que se disparó luego del anuncio y que a partir de la próxima entrega iban a ajustar su proceso de selección, exigiendo a los nominadores que notifiquen si están al tanto de algún comportamiento del candidato propuesto, que sea inconsistente con los valores de la ACM.[10]

Mientras ese es un buen primer paso, ¿es suficiente?, especialmente teniendo en cuenta todo el esfuerzo que la comunidad de las ciencias de la computación, y STEM (Science, Technology, Engineering, and Mathematics) en general, está haciendo para ampliar la participación a sectores que han estado tradicionalmente excluidos o subrepresentados. Posiblemente sea hora de repensar el rol que queremos que los valores éticos efectivamente ocupen dentro de nuestra comunidad. Abrazarlos seguramente requerirá un cambio cultural muy grande de las instituciones educativas y organizaciones científicas que ante este tipo de situaciones, deberán ir más allá de salvar su imagen, al menor costo posible.

Resulta sorprendentemente paradójico que un científico sostenidamente contrario a la diversidad e inclusión se haga acreedor de una premiación que conmemora el legado de Alan Turing, uno de los fundadores de la teoría de la computación quien tanta discriminación sufrió por su orientación sexual, habiendo sido procesado y condenado por ello. Tuvieron que pasar más de cincuenta años para que recibiera el indulto y perdón póstumo que tantos reclamaban. ¿Tendrán las víctimas de la discriminación por parte de Ullman la misma suerte? Solo el tiempo nos dará una respuesta…

7 https://csforinclusion.wordpress.com/.

8 https://www.acm.org/about-acm/mission-vision-values-goals.

9 https://twitter.com/TheOfficialACM/status/1379891090246004744.

10 https://www.acm.org/response-to-letter.