

Robin Milner: el padre de la teoría de la concurrencia

Tributo

Juan Francisco Díaz Frías Jhonatan Serna Michell Guzmán

Escuela de Ingeniería de Sistemas y Computación,
home page: <http://eisc.univalle.edu.co/>
Universidad del Valle, Cali, Colombia
jdiaz@univalle.edu.co

28 de Abril de 2010

Plan de la Parte I

- 1 Referencias
- 2 Evolución académica
 - Antes de la Universidad
 - Pregrado
 - Carrera desde la Graduación

Plan de la Parte I

- 1 Referencias
- 2 Evolución académica
 - Antes de la Universidad
 - Pregrado
 - Carrera desde la Graduación

Plan de la Parte II

- 3 Legados
 - Prueba automática de teoremas: LCF
 - Lenguaje de programación ML
 - Teoría de Sistemas Concurrentes (CCS, Pi)
 - Hennessy-Milner Logic
 - Hindley-Milner type inference
 - Otros
- 4 Píldoras de su filosofía ...

Plan de la Parte II

- 3 Legados
 - Prueba automática de teoremas: LCF
 - Lenguaje de programación ML
 - Teoría de Sistemas Concurrentes (CCS, Pi)
 - Hennessy-Milner Logic
 - Hindley-Milner type inference
 - Otros
- 4 Píldoras de su filosofía ...

Plan de la Parte III

5 Premios y distinciones

Parte I

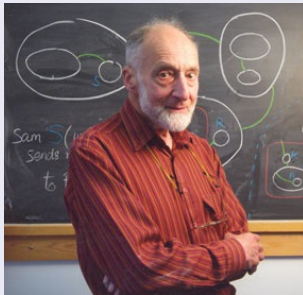
Referencias e historia

Antes de empezar ...

Mi objetivo es ...

...dejarlos con la motivación de conocer y estudiar en profundidad a una persona que marcó las Ciencias de la Computación:

Robin Milner



Referencias (1)

Entrevistas

- An interview with Robin Milner: Entrevista Martin Berger, 3 de Septiembre de 2003

Referencias (1)

Entrevistas

- *An interview with Robin Milner*: Entrevista Martin Berger, 3 de Septiembre de 2003

Referencias (2)

Sitios WEB

- Su sitio WEB personal
- Fairdene: Robin Milner
- guardian.co.uk Robin Milner obituary

Referencias (2)

Sitios WEB

- Su sitio WEB personal
- Fairdene: Robin Milner
- guardian.co.uk Robin Milner obituary

Referencias (2)

Sitios WEB

- Su sitio WEB personal
- Fairdene: Robin Milner
- guardian.co.uk Robin Milner obituary

Referencias (2)

Sitios WEB

- Su sitio WEB personal
- Fairdene: Robin Milner
- guardian.co.uk Robin Milner obituary

Plan

- 1 Referencias
- 2 Evolución académica
 - Antes de la Universidad
 - Pregrado
 - Carrera desde la Graduación

Antes de la Universidad

Orígenes

- Nació en 1934 en Yealmpton
- Su padre fue oficial del ejército
- A la edad de 8 años aprendió latín, y a los 10 griego
- En su bachillerato le encantaba la geometría proyectiva
- Realizó servicio militar de 1952 a 1954
- La música era su hobby. Tocaba el oboe.
- Tomó una beca para estudiar matemáticas en el King's College de Cambridge en 1954

Plan

1 Referencias

2 Evolución académica

- Antes de la Universidad
- **Pregrado**
- Carrera desde la Graduación

Pregrado(1)

Matemáticas en el King's College

- Empezó en segundo año
- En 1956 fué a un curso de verano sobre la máquina EDSAC, pero no le gustó programar
I regarded programming as really rather inelegant. You'd write one instruction after the other, and it was all rather arbitrary. It seemed to me that programming was not a very beautiful thing. So I resolved that I would never go near a computer in my life!
- En ese momento no conocía nada sobre la obra de Turing

Pregrado(2)

Cambio a Filosofía

- En el '56 se cambió a Filosofía

I also did a certain amount of analytical philosophy at that time. I got interested in and puzzled by Russell's theory of descriptions and names

- Cuando terminó sus estudios, estaba cansado del mundo académico y decidió irse de Cambridge

Plan

1 Referencias

2 Evolución académica

- Antes de la Universidad
- Pregrado
- Carrera desde la Graduación

Después de Cambridge

Después de la Graduación

- En 1957 va a Londres y realizó variedad de trabajos, entre otros, como profesor de matemáticas
- En 1961 ingresa a Ferranti como programador

I didn't move into it with great enthusiasm, but I realised that I should get a job.

- Escribe su primer compilador
- Decide buscar un trabajo en la academia y va a la City University en 1963
- Comenzó a interesarse por la IA
- Nunca hizo un doctorado

Después de Cambridge(2)

Trabajos en verificación

- En 1968 va a Swansea y trabaja como investigador.
- Conoce a David Cooper e ingresa a un pequeño grupo que trabaja sobre verificación y prueba automática de teoremas.
- En el '71 va a Stanford y allí desarrolla LCF, un probador asistido por máquina.
- En 1973 vuelve al Reino Unido como profesor de ciencias de la computación en la Universidad de Edimburgo.
- Allí creó ML.

Profesor en Edimburgo y Cambridge

Trabajos en Concurrencia

- En 1980 publicó una base matemática para entender los sistemas concurrentes llamada *Calculus for Communicating Systems (CCS)*.
- En 1986 fundó el *Laboratory for Foundations of Computer Science (LFCS)*.
- Allí lideró el desarrollo del Cálculo π .
- En 1995 entra a la Universidad de Cambridge como profesor de ciencias de la computación hasta 1999
- Sus investigaciones continúan tanto en Cambridge como en Edimburgo

Parte II

Legado académico y filosófico

Legados (1)

Cinco grandes áreas

- Prueba automática de teoremas: *LCF*
- Lenguaje de programación: *ML*
- Teoría de Sistemas Concurrentes: (*CCS, Pi*)
- Hennessy-Milner Logic
- Hindley-Milner type inference
- Otros

Plan

- 3 Legados
 - Prueba automática de teoremas: LCF
 - Lenguaje de programación ML
 - Teoría de Sistemas Concurrentes (CCS, Pi)
 - Hennessy-Milner Logic
 - Hindley-Milner type inference
 - Otros
- 4 Píldoras de su filosofía ...

Legados (2)

LCF (Logic for Computable Functions)

- LCF (Logic for Computable Functions), fue el nombre de Milner de la lógica ideada por Dana Scott en 1969, pero no publicada hasta 1993
- Es un sistema de razonamiento formal asistido por máquinas o computadoras
- Es diseñado para permitir al usuario interactivamente generar pruebas formales acerca de funciones computables y funcionales en una amplia variedad de dominios
- Este se extendió en una amplia variedad de versiones desarrolladas en distintas universidades como Stanford LCF, Edinburgh LCF, y Cambridge LCF.

Plan

- 3 Legados
 - Prueba automática de teoremas: LCF
 - **Lenguaje de programación ML**
 - Teoría de Sistemas Concurrentes (CCS, Pi)
 - Hennessy-Milner Logic
 - Hindley-Milner type inference
 - Otros
- 4 Píldoras de su filosofía ...

Legados (3)

Lenguaje de programación ML

- Lenguaje de programación de propósito general.
- Fue concebido para desarrollar técnicas de prueba en el probador de teoremas *LCF*
- Es conocido por su uso del algoritmo de inferencia de tipos Hindley-Milner
- Otras características: Multiparadigma, imperativo, multifuncional, utiliza tipamiento estático, inferido y fuerte.
- Algunos lenguajes de la familia ML: *Standard ML*, *Caml*

Plan

- 3 Legados
 - Prueba automática de teoremas: LCF
 - Lenguaje de programación ML
 - Teoría de Sistemas Concurrentes (CCS, Pi)
 - Hennessy-Milner Logic
 - Hindley-Milner type inference
 - Otros
- 4 Píldoras de su filosofía ...

Legados (4-A)

Teoría de Sistemas Concurrentes (*CCS, Pi*)

- La intención fue extender el alcance de la semántica denotacional mas allá de los programas de entrada/salida secuenciales.
- Aunque las redes de Petri existían desde 1960s, la cuestión elusiva fue cómo construir comportamientos complejos a partir de unos mas simples, en una forma modular
- Alrededor de 1980 desarrollo CCS (Calculus of communicating systems), uno de los primeros cálculos algebraicos para el análisis de sistemas concurrentes.

Legados (4-B)

Teoría de Sistemas Concurrentes (CCS, Pi)

- A finales de los años 80's desarrolló el cálculo π junto a Joachim Parrow y David Walker (basado en las ideas de Mogens Nielsen) como una continuación del cálculo CCS.
- El objetivo principal era el de describir computaciones concurrentes cuya configuración puede cambiar durante la computación
- Estos cálculos son parte de una búsqueda continua de una teoría que una la computación y la comunicación.

Plan

- 3 Legados
 - Prueba automática de teoremas: LCF
 - Lenguaje de programación ML
 - Teoría de Sistemas Concurrentes (CCS, Pi)
 - **Hennessy-Milner Logic**
 - Hindley-Milner type inference
 - Otros
- 4 Píldoras de su filosofía ...

Legados (4-C)

Hennessy-Milner Logic

- Es una lógica modal en ciencias de la computación.
- Esta es usada para especificar propiedades de un LTS (Sistema de transición etiquetado), una estructura similar a un autómata

Plan

- 3 Legados
 - Prueba automática de teoremas: LCF
 - Lenguaje de programación ML
 - Teoría de Sistemas Concurrentes (CCS, Pi)
 - Hennessy-Milner Logic
 - **Hindley-Milner type inference**
 - Otros
- 4 Píldoras de su filosofía ...

Legados (5)

Hindley-Milner type inference

- Inferencia de tipos, o tipamiento implícito, se refiere a la habilidad de deducir automáticamente el tipo de un valor en un lenguaje de programación.
- Es un algoritmo comúnmente usado para ejecutar inferencia de tipos.
- Este ha sido referenciado en el pasado como chequeo de tipo polimórficos o algoritmo W.
- El origen de este algoritmo es el algoritmo de inferencia de tipos para el cálculo λ simplemente tipado, el cual fue dividido por Haskell Curry y Robert Feys en 1958.

Plan

- 3 Legados
 - Prueba automática de teoremas: LCF
 - Lenguaje de programación ML
 - Teoría de Sistemas Concurrentes (CCS, Pi)
 - Hennessy-Milner Logic
 - Hindley-Milner type inference
 - Otros

- 4 Píldoras de su filosofía ...

En resumen . . .

- Ha escrito por lo menos 8 libros, y ha editado por lo menos 2, ha publicado mas de 60 artículos como único autor, mas de 20 artículos con otros autores y por lo menos 20 reportes que no han sido publicados.
- En sus últimos años estaba trabajando en un formalismo para Computación Ubicua que incluya el cálculo CCS y el cálculo Pi: los bigrafos.

Píldoras de su filosofía (1)

Relación entre lógica y computación

The idea of a machine proving theorems in logic, and the idea of using logic to understand what a machine was doing ... this double relationship to inspire me because it was clearly not very simple. The relationship wasn't just an inverse relationship "I bite you and you bite me": the computer says: "I will automate your logic" and the logic says "I will tell you what your programs means". These are two different relationships.

Píldoras de su filosofía (2)

Sobre simulación

Simulation intrigued me because it was about how you represent the real world inside a computer. I remember being puzzled by how to define the agents in a simulated process. For example if you have humans passing buckets down the chain, then you simulate the humans, but shouldn't you simulate the buckets in exactly the same way? From the bucket's point of view, the humans are moving past the bucket, from the human's point of view, the buckets are moving past the humans. So there was some extraordinary looseness about what could be meant by representing a real-world process.

Píldoras de su filosofía (3)

Los lenguajes para controlar computadoras

Every tool designed by man is a prosthetic device, and for every prosthetic device there is a means of control. Many tools are physical, and their control is manual. In contrast, computers are the most complex tools ever invented, and they are tools of the mind; the means of control is hardly muscular - it is primarily linguistic. Quite simply, the versatility of computers is exactly equal to the versatility of the languages by which we prescribe their behaviour, and this appears to be unbounded.

Píldoras de su filosofía (4)

Que es computación?

Computing is not only about a computer's internal action; it is about the way a computer behaves as part of a larger system, say an aircraft. It follows that the terms in which the computer behaviour is prescribed must harmonize with the way we describe information flow in such systems. Thus computing expands into informatics, the science of information and communication.

Píldoras de su filosofía (5)

No existe el lenguaje perfecto

We did not design the perfect language - that does not exist. But I believe that the Standard ML project sets a standard for how computing languages should be conceived; not just as syntactic conventions for assembling millions of lines of code, but as the expression of dynamic mathematical models.

Píldoras de su filosofía (6)

Ciencias de la computación

The term 'informatics' describes our discipline better than 'computer science', as it is now concerned with communicating and informing as well as with calculation. The most long-lived and famous informatic structure is the von Neumann machine, which gave rise to an impressive series of languages and theories. But the von Neumann machine treats primarily sequential computing on a single machine, and does not scale up to explain modern informatic behaviour.

Parte III

Premios y distinciones

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (1)

- 1987 British Computer Society Technical Award, for "The Programming Language ML"
- 1988 Fellow of the Royal Society of London
- 1988 Distinguished Fellow of the British Computer Society
- 1991 Turing Award winner
- 1993 Fellow of the Royal Society of Edinburgh Honorary Member of the IEEE
- 1994 Fellow of ACM Member of St. Petersburg Academy of Science
- 1994 ITALGAS prize, Turin
- 1994 F.L. Bauer prize, Munich, 1994.

Premios y distinciones (2)

- 2001 ACM SIGPLAN award for Achievement in Programming Languages
- 2001 UK Computing Research Committee
- 2004 Royal Medal, Royal Society of Edinburgh
- 2005 Distinguished Achievements Award, European Association for Theoretical Computer Science
- 2005 Foreign member, Académie Française des Sciences
- 2008 Foreign associate, National Academy of Engineering, USA.

Premios y distinciones (2)

2001 ACM SIGPLAN award for Achievement in Programming Languages

2001 UK Computing Research Committee

2004 Royal Medal, Royal Society of Edinburgh

2005 Distinguished Achievements Award, European Association for Theoretical Computer Science

2005 Foreign member, Académie Française des Sciences

2008 Foreign associate, National Academy of Engineering, USA.

Premios y distinciones (2)

2001 ACM SIGPLAN award for Achievement in Programming Languages

2001 UK Computing Research Committee

2004 Royal Medal, Royal Society of Edinburgh

2005 Distinguished Achievements Award, European Association for Theoretical Computer Science

2005 Foreign member, Académie Française des Sciences

2008 Foreign associate, National Academy of Engineering, USA.

Premios y distinciones (2)

- 2001 ACM SIGPLAN award for Achievement in Programming Languages
- 2001 UK Computing Research Committee
- 2004 Royal Medal, Royal Society of Edinburgh
- 2005 Distinguished Achievements Award, European Association for Theoretical Computer Science
- 2005 Foreign member, Académie Française des Sciences
- 2008 Foreign associate, National Academy of Engineering, USA.

Premios y distinciones (2)

- 2001 ACM SIGPLAN award for Achievement in Programming Languages
- 2001 UK Computing Research Committee
- 2004 Royal Medal, Royal Society of Edinburgh
- 2005 Distinguished Achievements Award, European Association for Theoretical Computer Science
- 2005 Foreign member, Académie Française des Sciences
- 2008 Foreign associate, National Academy of Engineering, USA.

Premios y distinciones (2)

- 2001 ACM SIGPLAN award for Achievement in Programming Languages
- 2001 UK Computing Research Committee
- 2004 Royal Medal, Royal Society of Edinburgh
- 2005 Distinguished Achievements Award, European Association for Theoretical Computer Science
- 2005 Foreign member, Académie Française des Sciences
- 2008 Foreign associate, National Academy of Engineering, USA.

Premios y distinciones (2)

- 2001 ACM SIGPLAN award for Achievement in Programming Languages
- 2001 UK Computing Research Committee
- 2004 Royal Medal, Royal Society of Edinburgh
- 2005 Distinguished Achievements Award, European Association for Theoretical Computer Science
- 2005 Foreign member, Académie Française des Sciences
- 2008 Foreign associate, National Academy of Engineering, USA.

Para terminar ...

- Muchas Gracias ...