

DE LOS SISTEMAS FORMALES A LOS SISTEMAS EVOLUTIVOS PASANDO POR EL TEOREMA DE GÖDEL

Esquema

29 abril 96, Octubre del 2009

Fernando Galindo Soria

1. Antecedentes históricos

Conjuntos de Cantor
Siglo XIX Física
Principios siglo XX
Problemas de Hilbert
1910 Principia Mathematica => Sistemas Formales
Geometría Euclidiana
=>1932 Axiomas de Peano (Sistema, Sistema Formal)
Geometrías no Euclidianas
Teorema de Gödel
Física de Newton => Física de Einstein => Mecánica Cuántica
Replanteamiento de la Matemática
Algoritmo
Problemas computables y no computables
Algoritmo recursivo
Lingüística Matemática
Sistemas de Reescritura
Replanteamiento general
La estructura de las Revoluciones Científicas Thomas Kuhn
Sistemas Evolutivos

1880 Físicos => Las leyes básicas de la física ya están descubiertas solo hay que aplicarlas
Principia Mathematica => 1910 Russel y Withehead
Las leyes básicas (principios) de la matemática ya están descubiertas solo hay que aplicarlas
Conjunto de axiomas, reglas, leyes que se aplican para resolver problemas de un Área
Conjunto de leyes básicas que se pueden aplicar para resolver problemas de un Área
Conjunto de axiomas, reglas, leyes fijo e inmutable que describen el comportamiento de un área
Por ejemplo en aritmética se pueden dar sistemas formales como la definición de un numero natural Peano

El 1 es un numero natural

Si n es un numero natural entonces $n + 1$ es un numero natural

1 pertenece a \mathbb{N}

Si n pertenece a \mathbb{N} => $n + 1$ pertenece a \mathbb{N}

Así como las reglas anteriores se puede tener un conjunto de reglas que describan a la aritmética

Sin embargo en 1900 Hilbert dio una lista de problemas a resolver durante el siglo XX (algunos todavía no se resuelven por ejemplo el ultimo teorema de Fermat

$5^2 = 2^2 + 3^2$, pregunta para todo n pertenece a \mathbb{N} existe a, b, c pertenecen a \mathbb{N} tal que $a^n = b^n + c^n$
es un problema de la aritmética que aun esta en proceso de solución.

Aparentemente por muy completas que sean las reglas que describen un área, existen problemas que no se pueden resolver usando esas reglas

⇒ 1982 Gödel

Gödel demuestra que la aritmética no se puede formalizar, aun más demuestra que si se tiene un sistema formal este es incompleto [EXPLICAR] o es inconsistente [EXPLICAR]

Thomas Kuhn

La estructura de las Revoluciones Científicas

La Física no tiene un conjunto de reglas fijo e inmutable (sistema formal) al contrario es un área en constante cambio, donde las reglas se mantienen vigentes durante un tiempo hasta que se encuentra otra regla que describe mejor el problema estudiado o son reglas que se van depurando poco a poco y cada vez describen mejor el problema.

Ciencia, Leyes ⇒ Disciplina, Paradigma

Lo mismo pasa en Matemáticas, Biología, etc.

Poner otros ejemplos ⇒

Sistema Evolutivo

Conjunto de reglas que describen el comportamiento de sistemas pero que se pueden cambiar con el fin de describir mejor al sistema o soportar los cambios en este ⇒

2. De los Sistemas Formales a los Sistemas Evolutivos pasando por el Teorema de Gödel

2.1 Sistemas Formales

Cuando un sistema aplica un conjunto de reglas o instrucciones para resolver un problema se dice que es un *Sistema Formal*, y a los métodos que construyen programas de este tipo se les conoce como métodos de programación deductiva o formal.

Un sistema formal tiene un conjunto de reglas que nos permiten resolver un problema. Cuando hacemos un programas en Basic, C o Java, lo que se esta haciendo es un sistema formal, se le esta diciendo a la computadora, si aplicas esta regla, o esta otra, vas a resolver el problema.

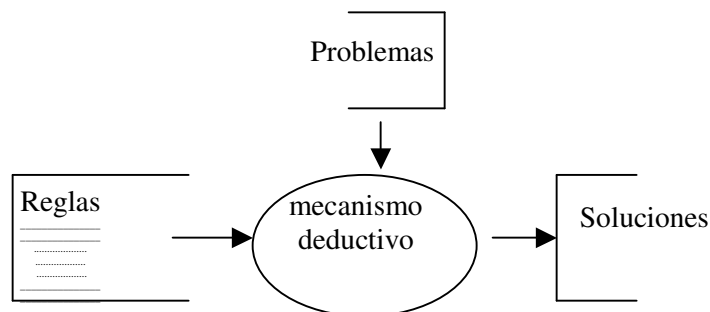


Figura 1. Sistema Formal

Los sistemas formales son los que mas se usan para programar, son muy usados y muy buenos, el problema surge cuando existe algún problema que no puedan resolver, por ejemplo, supóngase que hacemos un programa que reconoce la letra A, en el programa deberían de incluirse las características de la letra A, de modo que identificara A's similares, pero si se quiere reconocer una B no la reconocería, o tampoco reconocería la letra a minúscula, a cuadrada y otras A's incluso, a

menos que explícitamente se le diga como hacerlo, lo cual a la larga se vuelve muy pesado, porque siempre existe algo no cubierto por las reglas que tiene el sistema forma.

2.2 Sistemas que aprenden

El problema de la programación deductiva o de sistemas formales es que, se orienta a desarrollar sistemas fijos difíciles de modificar en tiempo real (ya que las modificaciones involucran reprogramar el sistema), por lo que, desde principios de los 60, se ha buscado desarrollar herramientas automatizadas capaces de obtener en forma automática el conjunto de reglas del sistema a partir de ejemplos y descripciones generales de un programa, en lo que se conoce como Machine Learning o maquinas o sistemas que aprenden.

A una herramienta o programa capaz de encontrar un conjunto de reglas a partir de ejemplos la conocemos como Mecanismo o *Herramienta Inductiva*; el esquema de una herramienta inductiva es el mostrado en la figura 1.

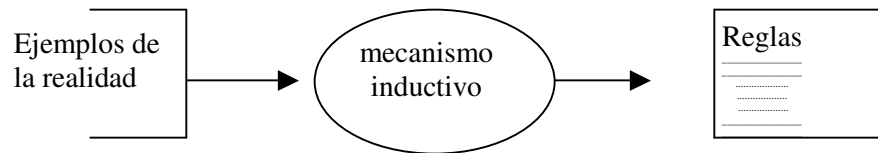


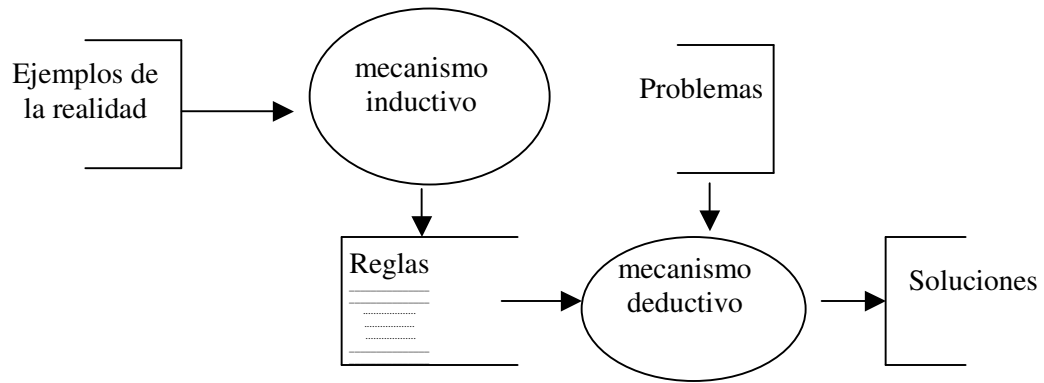
Figura 1. Herramienta inductiva

Lo anterior no significa que un sistema formal no sea útil, por el contrario nos han resuelto un gran número de problemas, por que la forma común de programar es mediante sistemas formales, hagamos de cuenta que los sistemas formales es como la teoría de Newton con respecto a la teoría de Einstein.

La física de Newton la usamos para muchas cosas, por ejemplo, cuando calculamos la velocidad de un proyectil no usamos la teoría de Einstein, con Newton de hace cientos de años nos funciona muy bien, si quiero ver como rebota una pelota, con Newton nos basta, si queremos saber el comportamiento de el proyectil cuando se acerca a la velocidad de la luz entonces ya entramos con Einstein; los sistemas formales son así, para espacios normales restringidos convencionales funcionan muy bien.

El problema es que ya no estamos en esos espacios, los problemas a los que ahora nos enfrentamos ya rebasan un poco a esa cotidianidad, los informáticos rompieron esa cotidianidad desde los años 50 con el surgimiento de la Inteligencia Artificial, cuando empezaron a enfrentar problemas que ya no eran tan fáciles de resolver utilizando sistemas formales, y que requerían de otro tipo de tratamiento, por ejemplo un problema de esa época era el reconocimiento de letras.

En los años 60 surge el área de Machine Learning (Sistemas que aprenden), la idea de una Machine Learning es que se invierte la situación, en lugar de que yo me preocupe de darle las reglas para que reconozca un objeto, le doy la capacidad de que reconozca o aprenda como esta hecho el objeto, los sistemas Machine Learning tienen un programa al cual por ejemplo le damos una A y el programa se las arregla para encontrar la estructura de esa A. Estos programas construyen un archivo o un sistema de ecuaciones dónde están representadas el conjunto de reglas que representan a esa A. Lo bonito de un Machine Learning es que uno le puede dar por ejemplo una A cuadrada, rectangular, o chiquita y va construyendo la representación de todas esas A's.



Learning Machines (Sistemas que aprenden) años 60

Los Machine Learning se empezaron a desarrollar desde los años 60, se le daba a la computadora un montón de ejemplos de aquello que quisiéramos que reconociera y encontraba las reglas que describían ese tipo de cosas o sistemas, y ya que tenían ese conjunto de reglas, se las pasaban a un programa formal al que por ejemplo se le daba una A cuadrada y entonces la computadora la reconocía como A cuadrada, lo que nos ahorrábamos era el trabajo de estar encontrando las reglas, El libro básico “Learning Machine” de Nilsson se escribió en 1965 .

El problema de los Machine Learning es que tiene dos grandes etapas: una “de aprendizaje” y otra Formal. Por lo que solo resuelve problemas que previamente se le enseñó a resolver, pero cuando llegan problemas que previamente no aprendió no los reconoce, si por ejemplo se le dan un montón de ejemplos de A's reconoce A's, pero si posteriormente se me ocurre colocar una A Itálica, ya no la reconoce si previamente no se le enseñó a reconocer ese tipo de A's.

El problema tanto de los sistemas formales como de los sistemas que aprenden o Machine Learning es que tarde o temprano les llegara un problema que no pueden resolver.

2.3 Teorema de Gödel

En 1930 Kurt Gödel estableció un teorema que lleva su nombre, el teorema de Gödel puede considerarse como un teorema fundamental en informática. El teorema dice en general que: un sistema Formal, es incompleto, o inconsistente, un sistema formal es incompleto cuando tarde o temprano, no importando el número de reglas que tenga, va a existir un problema que se supone debería poder resolver y que no puede resolver, o sea le faltan reglas para poderlo resolver, e inconsistente significa que tarde o temprano, aplicando un mismo grupo de reglas encuentra una solución verdadera y una falsa para el mismo problema.

Si se logra que un sistema formal sea completo será inconsistente y si es consistente entonces será incompleto.

2.4 Sistemas Evolutivos

Este tipo de problema nos mantuvo ocupados durante los años setenta y principios de los ochenta, observamos en aquellos años que, cuando se requiere hacer una modificación al código, muchas veces los programas tienen miles de instrucciones y resulta poco práctico y tedioso de realizar cambios en la programación por la complejidad de la codificación, y tomando en cuenta que se realizan actualizaciones y modificaciones frecuentemente se tiene que estar reestructurando continuamente, de donde salió la idea de hacer un sistema que se encargara de hacer las modificaciones al código automáticamente y asimismo que éste se modificara a si mismo.

De donde la idea sería utilizar un programa formal que contuviera las reglas por ejemplo para reconocer una A , y que cuando este no reconociera un tipo de especial de A, mandara llamar a otro tipo de programa que se encargara de aprender, lo que se convertiría posteriormente en el principio o la primera idea de lo que es un **sistema evolutivo o sea un programa que permanentemente esta aprendiendo y reestructurando sus reglas** de donde el programa no mandaría Fatal error o warnings, simple y sencillamente si el programa no sabe, aprende.

Entonces la primera idea fue introducirle un mecanismo de retroalimentación a un sistema que aprende obteniendo los primeros sistemas evolutivos: O sea que para contar con un Sistema Evolutivo, lo único que hace falta es dotar las herramientas con la capacidad de actualizar su imagen de la realidad en tiempo real, con lo que se tiene la arquitectura de la figura 3.

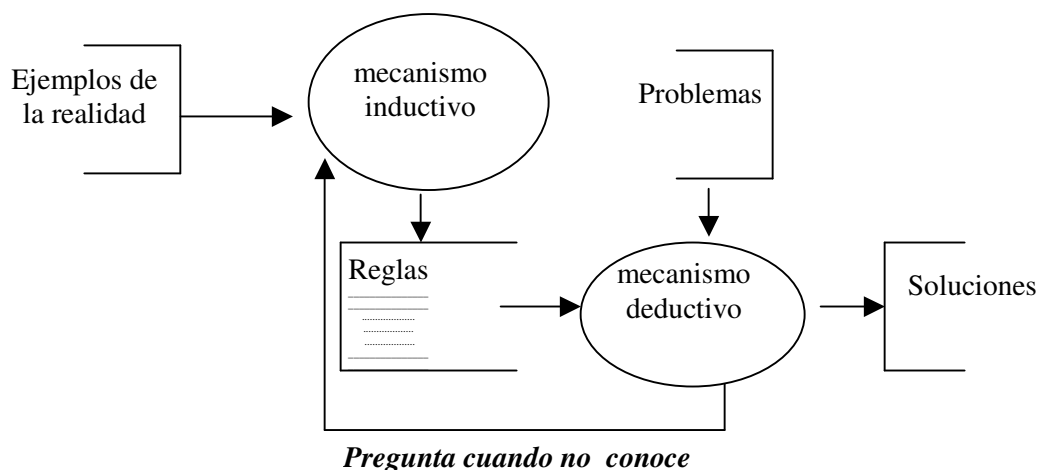


Figura 3. Sistema Evolutivo básico 1983

Generalizando se tiene la siguiente propuesta de Arquitectura de un Sistema Evolutivo en la figura 4.

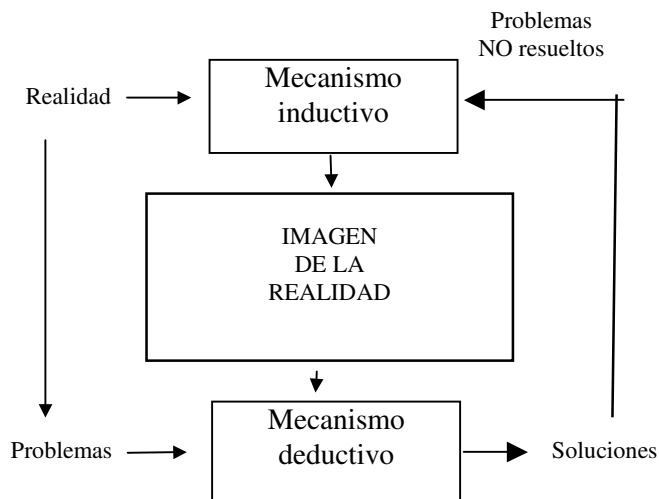


Figura 4. Propuesta de Arquitectura de un Sistema Evolutivo

A partir de esta propuesta se han desarrollado otras y en la actualidad se cuenta con varios enfoques de cómo construir este tipo de sistemas y en general la arquitectura de un Sistema Evolutivo. se puede ver constituida por tres grandes módulos interrelacionados.

- .La Representación de la Realidad.
- .El Manejador del Sistema Evolutivo.
- .Los Mecanismos de Interacción con el Ambiente.

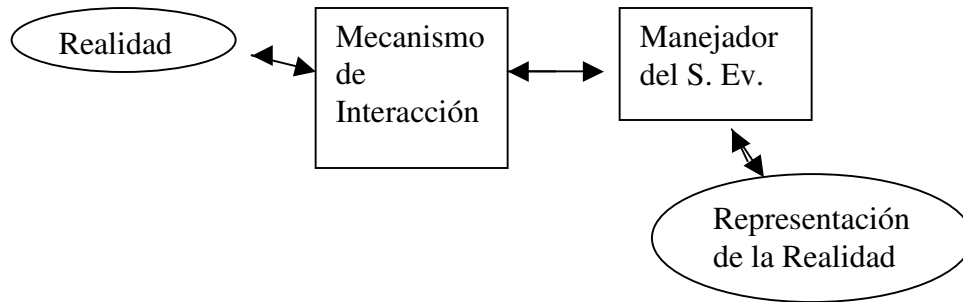


Figura 6. Arquitectura General de un Sistema Evolutivo

El módulo de *Representación de la Realidad* nos permite almacenar una imagen de una realidad dada en término de sus elementos, las relaciones entre éstos o estructura y su significado.

El *Manejador del Sistema Evolutivo*. es el mecanismo responsable de construir, mantener actualizada y aprovechar la Imagen de la Realidad a partir de los conocimientos y requerimientos detectados por el Mecanismo Interactivo.

Por su parte *el Mecanismo de Interacción* es el que permite que el sistema evolutivo se comunice con su ambiente ya sea para modificar su imagen de la realidad o llevar a cabo alguna acción sobre su exterior.