

A history of the history of programming languages.

- Mientras se celebraban los primeros 60 años del *ACM* se consideró importante hacer un reconocimiento sobre las contribuciones a la historia de los lenguajes de programación, mediante tres conferencias llamadas *ACM SIGPLAN History of Programming Language(HOPL)*. En 1978, 1993 y del 9 - 10 de Julio de 2007, considerando los lenguajes de programación más importantes que estuvieron involucrados en el desarrollo de más de otros 40 lenguajes.
 1. HOPL-I tomó como tema FORTRAN, el primer lenguaje en ser estandarizado, dada por John Backus.
 2. HOPL-II tomó como tema C, compartiendo las memorias de su desarrollo por Dennis Ritchie.
 3. HOPL-III tomó como tema C++, ampliando su historia comenzada en HOPL-II por Bjarne Stroustrup.

Y éste artículo pretende contarnos acerca de éstas conferencias y despertar curiosidad para seguir investigando.

- El primero en promover las conferencias HOPL fue Jean E. Sammet. Sammet formaba parte del CODASYL Language Structure Group en 1960 y también un miembro del USASI X3.4 Committee en lenguajes de programación. En 1961 Sammet reportó distintos artículos acerca de los lenguajes de programación, y el mayor de ellos fue su investigación, *PROGRAMMING LANGUAGES: History and Fundamentals*. El cuál fue el primer libro en tomar un enfoque sobre la historia y el desarrollo de los lenguajes de programación. Éstas actividades le otorgaron un lugar en la Universidad de California en Los Angeles, como maestro de educación continua. Mientras trabajaba ahí, Michael Melchinoff, dirigente del departamento de ciencias de la computación, le realizó unas preguntas a Sammet sobre la creación de COBOL: "¿Por qué COBOL tiene verbo COMPUTE además de ADD, SUBTRACT, MULTIPLY y DIVIDE" y "¿Por qué los nombres de datos en COBOL están limitados a 30 caracteres?". De lo cuál Sammet creyó conveniente la idea de realizar una conferencia donde una persona clave en la creación de lenguajes de programación hablará de cómo fue desarrollado. Con el apoyo del SIGPLAN y por John A.N, se llevó a cabo la idea.
- **La primera conferencia HOPL, Los Angeles, CA, 1978**

El propósito de HOPL fue examinar el desarrollo temprano de determinados lenguajes de programación, con un énfasis en los aspectos técnicos del diseño y creación del lenguaje. El Comité del Programa definió un conjunto de criterios para considerar un lenguaje dentro del evento. Los cuáles fueron: *El lenguaje debía estar creado, y en uso para 1967; seguir en uso en 1977; y tener una considerable influencia en el campo del cómputo*. Algunos aspectos que no se aplicaron a ciertos lenguajes fueron: su uso, la influencia en el diseño de lenguajes, su impacto en el entorno del cómputo, si era novedoso y único. Después de una larga discusión se decidieron incluir 13 lenguajes en el HOPL: ***FORTRAN I, II y III; ALGOL, LISP, COBOL, APT Language, JOVIAL, GPSS(General Purpouse Simulation System), SIMULA, BASIC, PL/I, SNOBOL, APL***.

Para realizar las conferencias el comité seleccionó 82 temas específicos para guiar a los autores a realizar un buen texto y atender hechos e ideas importantes.

- Background
 1. Hechos básicos acerca de la organización dle proyecto y la gente.
 2. Costos y Calendarios.
 3. Hechos básicos acerca de la Documentación.
 4. Lenguajes/Sistemas conocidos en ese tiempo.
 5. Propósitos y Usuarios dirigidos.
 6. Origen y Motivación
- Razonamiento del contenido del lenguaje
 1. Factores de Entorno
 2. Funciones que serían programadas
 3. Principios del diseño del lenguaje
 4. Definición del Lenguaje
 5. Conceptos acerca de otros lenguajes
 6. Influencia de factores no-técnicos
- Evaluación posterior
 1. Conferencia de objetivos
 2. Contribuciones a los lenguajes
 3. Errores o cambios no deseados
 4. Problemas
- Implicaciones para los lenguajes usados y futuros
 1. Influencia directa
 2. Influencia indirecta

Ésta primera conferencia se celebró del 1 - 3 Junio de 1978. Durante ésta conferencia Grace Murray Hoper, coder en Bureau del Ordnance Computation Project de Harvard; relató una historia sobre la evolución de los lenguajes de programación: desarrollando rutinas compartidas con la MARK I, desarrollando en octal para BINAC, y desarrollando el primer compilador para el lenguaje de programación A-O en 1952. Destacando dos aspectos importantes para los lenguajes de programación: *crear mejores programas con un menor esfuerzo y permitir a no-especialistas en el tema (como ingenieros o matemáticos) utilizar las computadoras para resolver problemas.*

– **HOPL II, Cambridge, MA, 1993**

Después de un par de años Sammet y Lee comenzaron la idea d eun segundo HOPL para expandir el conocimiento del primer evento y profundizar en los temas. En 1990 el SIGPLAN autorizó la segunda edición, dónde tentativamente se hablaría de la evolución de lenguajes, la historia de las características y conceptos de los lenguajes y las clases de lenguajes existentes. La HOPL-II fue celebrado del 20 - 23 de Abril de 1993, y tenía por título "Diseño de un lenguaje como diseño". Brooks comenzó observando que las metas y requerimientos cambiaban durante el proceso de diseño y se veían pobremente reflejado en los resultados del diseño. El entorno exterior también cambiaba, añadiendo resultados negativos. Y por lo tanto,

Brooks recomendó a los diseñadores: *"diseñar un lenguaje, no es sólo modificar uno; se debe estudiar y usar el diseño de otras personas, diseñar de arriba a abajo, conocer bien la aplicación e iterar el diseño con problemas independientes."*

– **HOPL III, San Diego, CA, 2007**

En 2004, Barbara Ryder, y Brent Hailpern, ambos dirigentes del SIGPLAN, comenzaron el proceso para planear el HOPL-III. Dónde cada publicación del HOPL debería detallar la temprana historia o evolución de específicos lenguajes de programación. Las ideas preliminares pudieran documentar por 1996 y los lenguajes en uso en 1998. Será celebrada en Junio del 9 - 16, 2007.

1. ¿Cual es la diferencia entre inducción y recursión?

La inducción es un razonamiento que permite demostrar una infinidad de proposiciones, o una proposición que depende de un parámetro que toma una infinidad de valores enteros y la recursión es la forma en la cual se especifica un proceso basado en su propia definición. En pocas palabras la inducción sirve para demostrar y la recursión sirve para definir.

2. Demuestra por induccion lo siguiente:

$$\forall n \in N, n + 0 = n$$

– Base:

Con $n = 0$, usamos la definición de suma ($0 + n = n$), entonces: $0 + 0 = 0$ ✓

– H.I.:

Supongamos que $\forall k \in N$ se cumple que $k + 0 = k$, con $0 \leq k \leq n$ ✓

– P.I.:

Por demostrar: $(n + 1) + 0 = (n + 1)$

Sabemos que $(n + 1) = Sn$, entonces: $(n + 1) + 0 = (Sn) + 0$, por la definicion de suma tenemos que: $1(Sn) + 0 = S(n + 0)$, usando la H.I. tenemos que $n + 0 = n$, entonces: $(Sn) + 0 = Sn$

$$\therefore (n + 1) + 0 = (n + 1) \quad \checkmark$$

$$\forall n \in N, Sn = n + 1$$

– Base:

Con $n = 0$, por la definición, tenemos: $S 0 = 1 = 0 + 1$ ✓

– H.I.:

Supongamos que $\forall k \in N$ se cumple que $Sk = k + 1$, con $0 \leq k \leq n$ ✓

– P.I.:

Por demostrar: $S(n + 1) = (n + 1) + 1$

Por la definición de suma, tenemos que: $S(n + m) = (Sn) + m$, entonces: $S(n + 1) = (Sn) + 1$, usando la H.I. tenemos: $(Sn) + 1 = (n + 1) + 1$

$$\therefore S(n + 1) = (n + 1) + 1 \quad \checkmark$$

3. Define recursivamente el producto de números naturales $*$: $N \rightarrow N \rightarrow N$

$$* : N \rightarrow N \rightarrow N$$

$$n * 0 = 0$$

$$n * (Sm) = n + (n * m)$$

4. Utilizando el inciso anterior, demuestra lo siguiente:

$$\forall n \in N, n * 0 = 0 * n = 0$$

– Base:

Con $n = 0$, por la definición de $(*)$, tenemos: $0 * 0 = 0$ ✓

– H.I.:

Supongamos que $\forall k \in N$ se cumple que $0 * k = 0$, con $0 \leq k \leq n$ ✓

– P.I.:

Por demostrar: $0 * (n + 1) = 0$

Sabemos que $(n + 1) = Sn$, entonces tenemos que $0 * (n + 1) = 0 * (Sn)$, usando la definición de $(*)$ tenemos que: $0 * (Sn) = 0 + (0 * n)$, usando la H.I. tenemos que $0 * n = 0$, entonces: $0 + (0 * n) = 0 + 0 = 0$

$\therefore 0 * (n + 1) = 0$ ✓

$\forall n \in N, n * 1 = 1 * n = n$

– Base:

Con $n = 0$, sabemos que: $0 * 1 = 0$ por la demostración anterior. ✓

– H.I.: Supongamos que $\forall k \in N$ se cumple que $k * 1 = k$, con $0 \leq k \leq n$ ✓

– P.I.:

Por demostrar: $(n + 1) * 1 = n + 1$

Sabemos que $(n + 1) = Sny1 = S0$, entonces $(n + 1) * 1 = (Sn) * (S0)$, usando la definición de $(*)$ tenemos: $(Sn) * (S0) = (Sn) + ((Sn) * 0)$, sabemos que $(Sn) * 0 = 0$ por la definición de $(*)$, entonces $(Sn) + ((Sn) * 0) = (Sn) + 0 = Sn = (n + 1)$

$\therefore (n + 1) * 1 = (n + 1)$ ✓

5. Enuncia el principio de inducción para listas.

Al igual que en la inducción sobre naturales, en la inducción sobre listas el caso base puede ser una lista con 0 elementos. En tal caso la hipótesis inductiva es $x.xs$, donde $\#xs \geq 0$.

6. Demuestra lo siguiente:

$\forall l \in ListA, l ++ nil = l$

– Base:

Con $l = nil$, sabemos que $nil ++ nil = nil$ por la definición de $++$. ✓

– H.I.:

Supongamos que $\forall l \in ListA$ se cumple que $l ++ nil = l$ ✓

– P.I.:

Por demostrar: $(x : l1) ++ nil = (x : l1)$ Por la definición de $(++)$ tenemos que:

$(x : l1) ++ nil = x : (l1 ++ nil)$,

usando la H.I. sabemos que $l1 ++ nil = l1$, entonces:

$x : (l1 ++ nil) = x : (l1) = x : l1$

$\therefore (x : l1) ++ nil = (x : l1)$ ✓