

Organización del Procesador

año 2016

Proyecto *Aritmética Intervalar*

Este trabajo deberá ser resuelto en grupos de **hasta 3** participantes. Los integrantes del grupo podrá distribuirse el trabajo pero **TODOS** deben conocer en profundidad parte del mismo y de manera general su totalidad. El mismo será evaluado y formará parte de la evaluación general del alumno (cada integrante) en la asignatura.

Introducción

La aritmética intervalar fue concebida por Ramon E. Moore en 1957 mientras se desempeñaba como empleado de la *Lockheed Missiles and Space Co. Inc* en un intento de lidiar con los errores de redondeo en computaciones matemáticas imponiendo un límite a los mismos. Debido a los errores de redondeo intrínsecos de finitud de la representación computación de números, en cálculos complejos un número se calcula bajo limitaciones que llevan a considerarlo como el número mas/menos un grado de error: $x \pm \epsilon$. Teniendo en cuenta esta apreciación Moore pensó que sería interesante para lograr su objetivo, pensar a un número como el *intervalo* que expresa dicha expresión $[x - \epsilon, x + \epsilon]$.

Sus aplicaciones fueron incrementándose a través de los años brindando apoyo a diferentes áreas de las Ciencias. Las principales aplicaciones fueron a cómputos estadísticos y más recientemente para la representación de información geométrica (imágenes, polígonos, etc.) donde el advenimiento del poder de cómputo Gráfico GPUs ha llevado a considerar esta representación incorporando nuevas técnicas para lograr algoritmos rápidos y robustos.

Notación y Operaciones

La notación general para expresar un intervalo (cerrado en este caso particular) es $[\text{lím.inf.}, \text{lím.sup.}]$. Para simplificar la descripción del comportamiento de las operaciones, se utiliza \underline{x} para hacer referencia a un número intervalar, donde \underline{x} alude al valor del límite inferior y \bar{x} al límite superior del número. Utilizando esta notación, las operaciones (algunas de ellas) sobre números intervalares se definen de la siguiente manera:

- **suma:** $\underline{x} + \underline{y} = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$
- **resta:** $\underline{x} - \underline{y} = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]$
- **multiplicación:** $\underline{x} * \underline{y} = [\min((\underline{x} * \underline{y}), (\underline{x} * \bar{y}), (\bar{x} * \underline{y}), (\bar{x} * \bar{y})), \max((\underline{x} * \underline{y}), (\underline{x} * \bar{y}), (\bar{x} * \underline{y}), (\bar{x} * \bar{y}))]$
- **inf** (ínfimo): $\inf(\underline{x}, \underline{y}) = [\max(\underline{x}, \underline{y}), \min(\bar{x}, \bar{y})]$ o vacío
- **sup** (supremo): $\sup(\underline{x}, \underline{y}) = [\min(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y})]$

Objetivos del proyecto

El objetivo del proyecto es desarrollar una biblioteca (simplificada y sólo con enteros) eficiente para trabajar con esta aritmética. Para ello se propone implementar en assembler la estructura

de un número intervalar en 32bits (16bits para cada extremo entero del número) y las operaciones mencionadas. Además desarrollar una API (Interfaz de Programación de Aplicaciones) en C, para simplificar el uso de la misma. Finalmente exponer una Test Suite (casos de prueba) que ejecute la utilización de la misma para diferentes casos.