

Propuesta de Trabajo Final

Evaluación y Derivación Automática mediante Números Duales

Luciano David Duarte
Legajo: D-4293/5

Tema del Proyecto

Implementación de un lenguaje para expresar funciones matemáticas en una variable real, que permita obtener tanto el valor de la función como el valor exacto de su derivada en un punto mediante el uso de *números duales*.

Introducción y Justificación

Los números duales son expresiones de la forma:

$$a + b\varepsilon, \quad \text{con} \quad \varepsilon^2 = 0,$$

donde a representa la parte real y b la parte dual. Esta estructura algebraica permite que, al evaluar una función f sobre un número dual $x + 1\varepsilon$, se obtenga:

$$f(x + 1\varepsilon) = f(x) + f'(x)\varepsilon,$$

resultando el valor de la derivada como parte del mismo proceso de evaluación. Esta técnica se conoce como *diferenciación automática* y es ampliamente utilizada en software científico y de aprendizaje automático.

Propiedades Algebraicas de los Números Duales

Un número dual se define como una expresión de la forma:

$$a + b\varepsilon, \quad a, b \in \mathbb{R}, \quad \varepsilon^2 = 0.$$

El componente a se denomina *parte real* y b *parte dual* o infinitesimal.

Suma

$$(a + b\varepsilon) + (c + d\varepsilon) = (a + c) + (b + d)\varepsilon$$

Producto Usando la condición de nilpotencia $\varepsilon^2 = 0$, se obtiene:

$$(a + b\varepsilon)(c + d\varepsilon) = ac + (ad + bc)\varepsilon$$

Elementos neutros e inversos

$$0 = 0 + 0\varepsilon, \quad c = 1c + 0\varepsilon$$

El inverso aditivo es:

$$-(a + b\varepsilon) = -a - b\varepsilon$$

y un número dual es invertible si y solo si $a \neq 0$. En ese caso:

$$(a + b\varepsilon)^{-1} = \frac{1}{a} - \frac{b}{a^2}\varepsilon$$

Evaluación de funciones Si f es diferenciable, la evaluación de la función sobre un número dual permite obtener automáticamente su derivada:

$$f(x + \varepsilon) = f(x) + f'(x)\varepsilon$$

La parte real de la evaluación es el valor de la función, mientras que la parte dual corresponde al valor exacto de su derivada.

Alcance funcional

- Reconocer expresiones en una variable x .
- Parser desde archivo con la gramática definida.
- Evaluación dual: cálculo simultáneo de valor y derivada en un punto.
- Operadores: $+, -, *, /, ^$
- Funciones unarias: $\sin, \cos, \tan, \exp, \log$
- Control de dominio y manejo de errores (división por cero, \log de no positivo, \tan en puntos singulares)

Expresiones válidas del lenguaje

Las expresiones aceptadas por el lenguaje están formadas por:

- **Constantes numéricas** (enteras o reales)
- **La variable x**
- **Operaciones aritméticas:**
 - Suma: $+$
 - Resta: $-$
 - Multiplicación: $*$
 - División: $/$
 - Potencia: $^$
- **Funciones unarias:**
 - Trigonométricas: $\sin(x), \cos(x), \tan(x)$
 - Exponencial: $\exp(x)$
 - Logarítmica: $\log(x)$
- **Paréntesis** para agrupar subexpresiones

Se respetará la precedencia usual:

$$\text{potencia} > \text{multiplicación/división} > \text{suma/resta}$$

y la asociatividad estándar de los operadores.

Ejemplos de expresiones válidas:

$3*x*x + 2*x + 1$
 $\sin(x)*\exp(x) - x^3$
 $(1 + \cos(x)) / (2*\log(x))$

Precedencias y asociatividad

- \wedge : prioridad más alta, asociatividad a la derecha
- $*$, $/$: prioridad media, asociatividad a la izquierda
- $+, -$: prioridad baja, asociatividad a la izquierda
- Paréntesis (...) tienen la máxima prioridad para agrupamiento

Manejo de dominio y errores

Se implementará un sistema de evaluación que devuelve un resultado marcado como `Either EvalError` a para distinguir éxito y fallos de dominio. Casos previstos:

- **División por cero:** al evaluar a/b , si el valor primal de b es (aproximadamente) cero, se retorna un error `DivByZero`.
- **Logaritmo:** $\log z$ exige $\text{primal}(z) > 0$; si no se cumple, se retorna `DomainError "log"`.
- **Tangente:** $\tan z$ tiene singularidades en $\frac{\pi}{2} + k\pi$. Se validará si $\cos(\text{primal}(z))$ es (aprox.) cero; en ese caso se retorna `DomainError "tan"`.
- **Potencias:** para exponentes no enteros y bases negativas se manejarán con las funciones de `Double` y se captarán `NaN/Infinity` como errores de evaluación.

La decisión sobre tolerancias numéricicas (comparación con cero) será documentada en el informe y parametrizable.

Interfaz de línea de comandos

Uso previsto:

```
dual archivo.expr x0
```

Salida:

```
f(x0)  = <valor>
f'(x0) = <derivada>
```

En caso de error de parseo o evaluación, se mostrará un mensaje explicativo.