Projeto da Linguagem de Programação

Lucas Virgili

Sumário

1 Fase 1

1.1 Domínio

1.1.1 Introdução

Há aproximadamente 2400 anos, Zeno de Elea abalou as fundações da matemática da época através da proposição de diversos paradoxos. Um deles é muito famoso:

> Um corredor nunca pode terminar uma corrida, já que para isso, ele primeiro tem que andar metade do percurso, em seguida um quarto, depois um oitavo, e assim por diante, *ad infinitum*.

Após 2000 anos, matemáticos dos séculos XII e XIII deram início à teoria de séries infinitas. Nessa teoria, a noção usual de soma, válida para conjuntos finitos, é expandida para coleções infinitas.

Dessa forma, o "paradoxo" de Zeno foi selecionado, já que, naquele contexto, a soma que representa as "etapas" que o corredor deve percorrer é conhecida

$$\frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^n} + \ldots \tag{1}$$

e seu resultado é 1.

1.1.2 Descrição do domínio

A teoria de séries e sequências não tem seu uso limitado a mostrar como gregos mortos estavam errados; ela é extremamente usada. Por exemplo, em análise, utiliza-se sequências de funções para demonstrar os teoremas de convergência monótona de Lesbegue. Esse teorema é importantíssimo em probabilidade, por exemplo.

Definimos sequências e séries abaixo.

Sequência: Uma função f cujo domínio é o conjunto dos inteiros positivos $1, 2, 3, \ldots$ é uma sequência infinita. O valor $f(n)$ é o enésimo *termo* da sequência.

Série: Dada uma sequência, podemos gerar uma nova sequência somando termos sucessivos. Logo, se temos uma sequência

$$a_1, a_2, \ldots, a_n, \ldots \tag{2}$$

Podemos gerar as seguintes "somas parciais":

$$s_1 = a_1, s_2 = a_1 + a_2, s_3 = a_1 + a_2 + a_3 \tag{3}$$

e assim continuarmos até a enésima soma parcial:

$$s_n = a_1 + a_2 + a_3 + a_4 + \ldots + a_n = \sum_{i=1}^{n} a_i \tag{4}$$

A sequência s_n das somas parciais é chamada de *série infinita* ou, simplesmente *série*, e é denotada por

$$a_1 + a_2 + a_3 + \ldots, \text{ ou } \sum_{i=1}^{\infty} a_i \tag{5}$$

Informalmente, dizemos que uma sequência converge se existe uma quantidade L para a qual a sequência se aproxima o quanto quisermos[1]. Uma série, então, converge se sua sequência s_n converge.

É comum, enquanto estamos trabalhando com sequências ou séries, escrevermos programas em uma linguagem como C para avaliar se uma sequência ou série converge.

1.2 Proposta da linguagem de programação

Para este projeto, propomos desenvolver uma linguagem de programação que permita a declaração de sequências e séries, bem como analisar a convergência das mesmas através de métodos conhecidos.

Por exemplo, será possível para o programador definir uma sequência e operar sobre elas:

[1]Formalmente, se para qualquer $\epsilon > 0$, existe um um número positivo N tal que $|f(n) - L| < \epsilon$ para qualquer $n \geq N$.

```
seq s
s(n) = 1 / n ^ 2
s(3) = 0.125
series(s, 3) = 0.875 ## calcula s_3
sequence_converges(s, 0.000001)
## -> (true, 0)
## Se a sequencia converge com uma precisao de 10 ^ (-5)


series_converges(s, 0.00001)
## -> (true, 1)
## Se a serie converge com uma "precisao" de 10 ^ (-5)
```

1.3 Elementos essencias à linguagem

Os seguintes elementos são fundamentais para a linguagem:

1. **Declarações**
   Nesta linguagem, as "variáveis" serão as sequências. Como visto no exemplo acima, o programador poderá declarar sequências utilizando a palavra reservada **seq**.

2. **Operadores**
   Sejam a_n e b_n duas sequências convergentes. É fácil mostrar que a série

   $$\sum_{n=1}^{\infty}(\alpha a_n + \beta b_n E) \tag{6}$$

   também converge e seu limite é dado por

   $$\alpha \sum_{n=1}^{\infty} a_n + \beta \sum_{n=1}^{\infty} b_n \tag{7}$$

   Assim, podemos multiplicar séries convergentes por constantes numéricas e também podemos somar e subtrair séries convergentes. Logo, a linguagem irá fornecer os operadores soma $(+)$ e subtração $(-)$ entre séries e o operador produto $(*)$ entre uma constante e uma série.

3. **Funções**
   A linguagem oferecerá ao programador as seguintes funções:

Função	O que ela calcula
series(sequencia, n)	calcula a enésima soma parcial de sequencia
sequence_converges(sequenca, precisao)	diz se sequência converge com precisão precisao
series_converges(serie, precisao)	diz se a série converge com precisão precisao

2 Fase 2

2.1 Tipos de dados

2.1.1 Tipos primitivos

2.1.2 Tipos compostos

2.2 Expressões

2.3 Comandos

2.4 Vinculação

2.5 Sistema de tipos