Introdução a uma Análise Intervalar Computável

Aarão Lyra[1] **e Benjamin R. Callejas Bedregal**[2]

1 Introdução

A qualidade do resultado em computação científica depende do conhecimento e controle dos erros na computação. Algoritmos convencionais, chamados algoritmos pontuais, computam uma resposta e às vezes uma estimativa do erro. No entanto, o usuário não pode conseguir uma resposta exata sem o auxílio de um análise rigorosa dos erros, o qual é extensa, dispendiosa e nem sempre viável. Por outro lado, técnicas intervalares podem ser programadas em computadores de tal modo que a computação possua uma rigorosa e completa análise do erro no resultado.

A matemática intervalar é uma teoria originada na década de 60 com o objetivo de responder questões de exatidão e eficiência que surgem na prática da computação científica e na resolução de problemas numéricos. Em [Aci91] é introduzido o conceito de espaço dos reais parciais (intervalos de extremos reais) munida da sua topologia de Scott cujos elementos são vistos como intervalos de informação. Cada elemento do espaço informa sobre um objeto real ou intervalo do espaço de Moore. Uma conseqüência desta mudança é a compatibilidade entre as propriedades da inclusão-monotonicidade e sua topologia de Scott.

As abordagens clássicas para teoria da computabilidade tratam com problemas discretos (por exemplo, sobre os números naturais, números inteiros, strings sobre um alfabeto finito, ou sobre grafos, etc.). No entanto, campos da matemática pura e aplicada tratam com problemas envolvendo números reais, números complexos. Isto acontece, por exemplo, em análise numérica, sistemas dinâmicos, geometria computacional e teoria da otimização. Assim, uma abordagem computacional para problemas contínuos é desejável, ou ainda necessária, para tratar formalmente com computações analógicas e em computações científicas em geral.

A computabilidade sobre conjuntos contáveis é obtida a partir do desenvolvimento de uma teoria da computabilidade sobre os números naturais [BL74] de tal modo que nos outros conjuntos se reduz à computabilidade no conjunto dos números naturais. Como cada número real parcial pode ser visto como limite de uma seqüência convergente de intervalos racionais e inversamente, podemos esperar que os números reais parciais tenham um papel numa teoria da computabilidade para conjuntos com a cardinalidade do contínuo semelhante a dos números naturais na teoria da computabilidade para conjuntos contáveis. Na literatura, existem diferentes abordagens para a computabilidade nos números reais, mas, uma importante diferença entre estas abordagens, está na maneira como é representado o número real [Gia93]. Num dos primeiros artigos sobre teoria dos domínios, D.Scott [Sco70] sugeriu que o cpo consistindo de intervalos da reta real poderia ser usado para estudar computabilidade sobre os números reais. Os espaço, em teoria dos domínios, para estudar computabilidade são chamados de domínios (contínuos) efetivamente dados. Previamente Martin-Löf [ML70] construiu um espaço similar de

[1] Mestre em Sistemas e Computação pela UFRN

[2] Doutor em Ciências da Computação pela UFPE, Professor da UFRN

aproximações. Em ambos os casos a reta real é mergulhada no espaço de aproximações onde a noção de computabilidade pode ser definida de uma maneira natural. Muitos resultados concernentes à teoria da computabilidade sobre os números reais, podem ser obtidos neste contexto. Nesta abordagem, uma aproximação da saída com precisão arbitrária é computada a partir de uma aproximação razoável da entrada [Bra95]. Uma outra linha de pesquisa para computabilidade real foi desenvolvida por Blum, Shub e Smale [BSS89]. Nesta aproximação, as chamadas máquinas BSS, um número real é visto como uma entidade acabada e as funções computáveis são geradas a partir de uma classe de funções básicas (numa maneira similar as funções parciais recursivas). Mesmo sabendo que cada uma destas aproximações tem seus méritos, nenhuma tem sido aceita pela maioria dos matemáticos e cientistas da computação.

Neste trabalho, estudamos o modelo BSS, desenvolvida por Blum, Shub e Smale [BSS89], usado para se caracterizar uma teoria da computabilidade sobre os números reais e extendemos este para se modelar a computabilidade no espaço dos intervalos. Nesta aproximação, um número real é visto como uma entidade acabada e as funções computáveis são geradas a partir de uma classe de funções básicas (numa maneira similar as funções parciais recursivas). Veremos, também uma aproximação para computabilidade intervalar epistemologicamente diferente desta, na qual um intervalo real é visto como o limite de intervalos racionais, e a computabilidade de uma função intervalar real depende da computabilidade de uma função sobre os intervalos racionais, isto é, ela se baseia no princípio de que podemos computar uma aproximação da saída com precisão arbitrária a partir de uma aproximação razoável da entrada.

2 Computabilidade Church-Turing

Existem várias "definições" do conceito de procedimento efetivamente dado, entretanto, devido sua natureza intuitiva, a maioria delas apelam para a intuição do leitor. Uma dessas definições tem a seguinte expressão: um procedimento efetivo é uma descrição finita, sem ambiguidade, de um conjunto finito de operações. As operações devem ser efetivas, no sentido de que existe um procedimento estritamente mecânico para computá-las [BrL74]. Uma função $f : A \longrightarrow B$ é efetivamente computável (aqui **Church-Turing computável**) se existe um procedimento efetivo tal que dado $a \in A$, ele pára, com saída $f(a)$, quando $a \in dom\ f$, e não pára, nos outro casos.

Em 1936, e nos anos subsequentes, foram apresentados vários modelos matemáticos de procedimento efetivo, posteriormente provados equivalentes em poder computacional. Devido a natureza não matemática das definições de procedimento efetivo, elas são conhecidas como evidência da tese de Church. Alguns dos mais importantes desses modelos são: funções λ-definíveis, por A. Church, funções recursivas parciais, por Kleene, máquinas de Turing, por A. Turing, Linguagem de programação PL, Algorítmos de Markov e sistemas de Post.

Informalmente, dizemos que um procedimento efetivo computa uma função $f : A \longrightarrow B$ se e somente se ele produz conhecimento de $f(x)$ a partir do conhecimento de x. A noção de conhecimento depende da natureza dos conjuntos A e B [Esc92]. Os modelos enunciados, acima, assumem que o conhecimento dos conjuntos A e B é expresso finitamente, como uma palavra de algum alfabeto finito.

Seja Σ um conjunto finito não vazio de símbolos ou letras (um alfabeto). Uma palavra sobre Σ é qualquer cadeia finita de símbolos de Σ. O conjunto de palavras sobre Σ, incluindo a palavra vazia, $\flat$, será denotada por Σ^* . O produto Cartesiano n-ário de Σ^* $(\underbrace{\Sigma^* \times \cdots \times \Sigma^*}_{n-vezes})$ será denotado por Σ^{*n}.

No sentido de caracterizar as funções efetivamente computáveis (aqui denominadas funções Church-Turing-computáveis), descreveremos um desses formalismos: as funções recursivas parciais.

2.1 Funções parciais recursivas

As funções iniciais usadas para definir as funções recursivas parciais são.

- A função projeção: $\pi : \Sigma^* \longrightarrow \Sigma^{*0}$, definida por $\pi(x) = \flat$
- A função identidade: $\iota : \Sigma^* \longrightarrow \Sigma^*$, definida por $\iota(x) = x$
- A função zero: $\zeta : \Sigma^{*0} \longrightarrow \Sigma^*$, definida por $\zeta(\flat) = 0$
- A função sucessor: $\varsigma : \Sigma^* \longrightarrow \Sigma^*$, definida por $\varsigma(x) = x + 1$

No que segue, definiremos algumas operações sobre funções. Observe que a última dessas operações pode não ser uma função total.

- Composição: $f \circ g(x) = f(g(x))$, onde $f : \Sigma^{*s} \longrightarrow \Sigma^{*t}$ e $g : \Sigma^{*r} \longrightarrow \Sigma^{*s}$.
- Combinação: $f \times g(x, y) = (f(x), g(y))$, onde $f : \Sigma^{*r} \longrightarrow \Sigma^{*s}$ e $g : \Sigma^{*t} \longrightarrow \Sigma^{*u}$.
- Exponenciação: $f^{\#}(x, y) = \underbrace{f \circ \cdots \circ f}_{y-vezes}(x)$, onde $f : \Sigma^{*r} \longrightarrow \Sigma^{*r}$.
- Repetição: $f^{\nabla}(x, y) = z$, para algum $z \in \Sigma^{*r}$ e para o menor valor n tal que $f^n(x, y) = (z, \varsigma(0))$, onde $f : \Sigma^{*r+1} \longrightarrow \Sigma^{*r+1}$,.

Definição 1 (BrL74) *Uma função $f : \Sigma^{*r} \longrightarrow \Sigma^{*s}$ é* **recursiva parcial** *sobre Σ se ela é uma das funções π, ι, ζ, ς em Σ ou pode ser obtida dessas funções por um número finito de composições, combinações, exponenciações ou repetições. Uma função recursiva parcial f sobre Σ é* **primitiva recursiva** *sobre Σ se ela pode ser obtida como acima, mas sem usar repetições. Uma função f é* **recursiva** *sobre Σ se ela é recursiva parcial, sobre Σ, e total (dom $f = \Sigma^{*r}$).*

Para fixar o significado das funções zero e sucessor, assim como das operações de exponenciação e repetição, assumiremos que $\Sigma^* = \mathbb{N}$.

Tese de Church: uma função $f : A \longrightarrow B$ é Church-Turing computável se e somente se existe uma função recursiva parcial $\rho : \mathbb{N} \longrightarrow \mathbb{N}$ (ou, equivalentemente, uma máquina de Turing que computa a função ρ) e funções de codificação[3] $\varepsilon_A : A \longrightarrow \mathbb{N}$ e $\varepsilon_B : B \longrightarrow \mathbb{N}$ de A e B, respectivamente, tal que o seguinte diagrama comuta.

Definição 2 (Rog67) *Seja A um conjunto. A diz-se* **recursivamente enumerável** *(r.e.) se existe uma codificação de A, $\varepsilon : A \longrightarrow \mathbb{N}$, e uma função recursiva parcial, $\rho : \mathbb{N} \longrightarrow \mathbb{N}$, tal que $Im(\varepsilon) = Im(\rho)$.*

Proposição 1 (BrL74) *Um conjunto A é r.e. se e somente se existe um procedimento efetivo que o enumera* ■

Proposição 2 (BrL74) *Uma função $f : A \longrightarrow B$ é Church-Turing-computável se e somente se $graf(f) = \{(a,b) \in A \times B : f(a) = b\}$ é r.e.* ■

Proposição 3 (BrL74) *Se A e B são conjuntos r.e., então $A \cap B$, $A \cup B$ e $A \times B$ também são r.e.* ■

3 Computabilidade no espaço dos números reais

3.1 Aproximação clássica

A noção de Church-Turing computabilidade é definida sobre conjuntos r.e.. Entretanto, a noção original de procedimento efetivo não se limita a conjuntos r.e.. Por exemplo, é natural pensar que a função identidade, em $\mathbb{R}$, seja "computável" num sentido mais amplo, onde a computação não seja realizada, necessariamente, num tempo finito.

Existem números reais que não podem ser representados como cadeias finitas. No entanto, exitem procedimentos mecânicos para obtê-los. Por exemplo, o número real e é o limite quando n tende para ∞ da sequência

$$S_n = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

Desde que, a soma, o produto, e a divisão sobre números racionais são funções computáveis, podemos assegurar que existe uma máquina de Turing que computa e. Assim, é possível definir uma noção de computabilidade sobre os números reais. A primeira noção de computabilidade sobre os reais foi introduzida por Turing em [Tur37]. Segundo Turing, um número real é computável, aqui chamado Turing-computável, se é possível representá-lo na forma

$$(2i-1)n + \Sigma_{r=1}^{\infty}(2c_r - 1)(\frac{2}{3})^r$$

[3] Uma codificação de A é uma função injetiva de A em $\mathbb{N}$ tal que ela própria é dada por um algoritmo e é reversível, no seguinte sentido: existe um algoritmo para reconhecer os números codificados e "decodifica-los" em A [Rog67].

onde a sequência binária $i\underbrace{||\cdots|}_{n-vezes}0c_1c_2\cdots c_r\cdots$ é computável por uma máquina de Turing. Em [Min67] existe uma definição equivalente, e mais intuitiva, de número real computável. Nesse trabalho, Minsky define um número real computável de forma análoga à maneira como é definida sequência de Cauchy, isto é, como uma sequência de dígitos interpretados com frações decimais, com a condição desses dígitos serem gerados sequencialmente por uma máquina de Turing. Existe uma outra noção de número real computável que apela para funções recursivas, em vez de máquinas de Turing: um número real, r, $(0 < r < 1)$ é computável se existe uma função recursiva, ρ, tal que $\mid r - \frac{\rho(n)}{n} \mid < \frac{1}{n}$, para cada $n \in \mathbb{N}$ [Mos57]. Em [Gia93,Gia93b] podem ser encontradas cinco definições diferentes de número real computável. Entretanto, a definição mais próxima com aquela que veremos neste capítulo, foi dada por Martin-Löf em [Lof70], onde para ele um número real é computável se o conjunto das aproximações (intervalos racionais) é r.e. Podemos reduzir esta definição àquela que considera duas sequências r.e. de números racionais, uma crescente, e outra decrescente, que convergem, ambas, para o mesmo número real.

Proposição 4 *Um número real x é computável se e somente se existem sequências, r.e., $p_0 < p_1 < p_2 < \cdots$ e $q_0 > q_1 > q_2 > \cdots$, de números racionais tais que*
$\lim_{i\to\infty} p_i = \lim_{i\to\infty} q_i = x$

Demonstração: Se x é computável, então, pela definição de Martin-Löf, o conjunto de intervalos $\{[p,q] : \ p < x < q\}$ é r.e. Logo, podemos obter sequências, r.e., $p_0 < p_1 < \cdots$ e $q_0 > q_1 > q_2 > \cdots$ de números racionais tais que os respectivos limites é o mesmo número real x.

Por outro lado, se $p_0 < p_1 < p_2 < \cdots$ e $q_o > q_1 > q_2 > \cdots$ são sequências de números racionais, r.e., convergindo para x, então, porque o conjunto de números racionais é r.e., podemos, sempre, decidir quando um número racional é maior do que um q_j ou menor do que p_i . Portanto, os conjuntos $\{p \in \mathbb{Q} : \ p < x\}$ e $\{q \in \mathbb{Q} : \ q > x\}$ são r.e., e consequentemente, o conjunto de intervalos, $\{[p,q] : \ p < x < q\}$, é r.e. ■

Proposição 5 (Min67) *Seja $\mathbb{R}_c$ o conjunto de números reais computáveis.*

1. *$\mathbb{R}_c$ é contável (portanto, $\mathbb{R}_c \neq \mathbb{R}$)*
2. *$\mathbb{R}_c$ não é r.e.*
3. *não é decidível se, dados dois números reais quaisquer, r e s, em $\mathbb{R}_c$, $r = s$ ou $r \neq s$.* ■

Em [Grz57] são apresentadas várias noções de computabilidade de funções sobre os reais. Uma delas propõe a seguinte definição: uma função $f : \mathbb{R} \longrightarrow \mathbb{R}$ é computável se existe uma função recursiva $\rho : \mathbb{N} \longrightarrow \mathbb{N}$ e um funcional computável, Φ, tal que, para cada $r \in \mathbb{R}$

$$\text{Se } \mid \frac{\rho(k)}{k+1} - r \mid < \frac{1}{k+1} \text{ para cada } k \in \mathbb{N}, \text{ então } \mid \frac{\Phi(f)(k)}{k+1} - f(r) \mid < \frac{1}{k+1}, \text{ para cada } k \in \mathbb{N}$$

Observe que a hipótese e a conclusão, dessa definição, são similares àquelas, da definição de número real computável, dado por [Mos57] (descrita acima). Segundo esta definição, uma função real computável, essencialmente, envia números reais computáveis em números reais computáveis.

3.2 O Modelo BSS

Em 1986 foi introduzido o conceito de um modelo de máquina que trabalha basicamente sob um anel ordenado, esse modelo foi definido por Blum, Shub e Smale tendo sido observado outras literaturas sobre a teoria da computação e complexidade. Essa máquina ficou sendo conhecida como BSS em homenagem aos seus criadores.

Seja A um anel ordenado, $l, m \in N$ e I=A^l, O=A^m, S=$\mathbb{Z}^+ \times \mathbb{Z}^+ \times A^\infty$ (entrada, saída e estado de possíveis configurações , respectivamente) onde $\mathbb{Z}^+$ é o conjunto dos números inteiros positivos. Uma máquina BSS M sobre A, com entrada I e saída O, é um grafo dirigido conexo com os nós sendo $N = \{1, \cdots, v\}$ para $v > 1$.

Estes nós são divididos em cinco categorias:

Nó-entrada: os arcos de entrada no nó v é zero (não chega nenhum arco) e possui apenas um arco de saída.

Nó saída: os arcos de saída no nó v é zero.

Nó-escolha: neste caso, o nó v tem dois arcos de saída e seus sucessores são $\beta^-(v)$ ou $\beta^+(v)$.

Em um nó escolha é feito uma tomada de decisão para qual nó seguirá: β^+ ou β^-.

Nó-computacional: neste caso, o nó v tem um arco de saída e tem associado uma função polinomial $g_v : S \to$ S (g_v pode ser racional se A é um corpo), ele é preciso quando $g_v(i, j, x) = (g_v^1(i), g_v^2(j), g_v^3(x))$, onde $g_v^1(i) \in \{1, i+1\}$ e $g_v^2(j) \in \{1, j+1\}$ $g_v^3(x)$ trata-se da função propriamente dita.

Nó quinta classe: Um quinto nó (tem um arco único de saída, e o próximo nó é β semelhante a um nó computacional) operando sobre este transformando-o no elemento $(i, j, x_1, x_2, \cdots, x_{j-1}, x_i, , x_{j+1}, \cdots) \in S$, isto é, x_i substitui x_j na j-ésima posição em A^∞. Não são feitas outras trocas no quinto nó. Se v representa este nó então esta função será escrita da seguinte forma:

$$g_v : S \to S.$$

Discutiremos agora as questões sobre as modificações necessárias na máquina de dimensão finita, nós utilizaremos $I = A^l, O = A^m$, ao mesmo tempo com um grafo dirigido (finito).

Os quatro primeiros nós são iguais diante de um grafo teórico, mas a definição de funções associadas deve ser extendida para ocultar o caso de dimensão infinita. Nós temos definido uma função polinomial $A^\infty \to A^\infty$. Mas este não afetará x_i para um i suficientemente grande na expressão $X = (x_1, x_2, \cdots) \in A^\infty$. Esta consideração forçou a introdução de outro tipo de nó dentro da máquina que foi chamado de quinto nó. O quinto nó é usado para acessar o x_i com o i suficientemente grande. Para estes nós foi utilizado $S = \mathbb{Z}^+ \times \mathbb{Z}^+ \times A^\infty$ com um elemento típico $(i, j, x_1, x_2, \cdots), i, j$ inteiros positivos.

É interessante observar que uma máquina BSS M é similar a uma máquina de acesso randômico clássica tendo um número infinito de registro e com capacidade de manter um elemento de A. Ele usa alguma entrada de A^l, dá alguma saída em A^m e pode ter um desempenho computacional (observada a complexidade) a nível polinomial (ou racional) envolvendo unicamente um número finito de registros. Esta máquina M também pode ter um desempenho simples se confrontar e executar uma instrução "escolha" dependendo do resultado da comparação.

Os dois registros adicionais inteiros são postos para permitir uma computação uniforme no caso que espaço de entrada é A^{∞}; de fato, sendo k_M a dimensão de M (o máximo entre as dimensões destas funções associadas com os nós computacionais de M).

Se o "quinto-nó" não é admitido, a máquina BSS M apenas não poderá usar nem modificar variáveis com indexação acima de k_M. Desta forma a entrada "ativa" é neste caso limitado por k_M.

Se quisermos uma máquina capaz de fazer computações que dependam da dimensão da entrada, nós temos que fazer em um nível não polinomial. Esta é a forma de adicionar dois registros inteiros (i e j). Eles são classificados em dois registros ordinários que podem ser incrementados ou reinicializados pelas instruções de computação ; quando o quinto nó é atingido o conteúdo do registro i-ésimo (A-valorado) é copiado para dentro do registro j-ésimo. Estes caminhos não são atualmente limitados pela dimensão da máquina, e assim, computações uniformes acima das ilimitadas são permitidas para receber espaço.

Nós podemos dar a notação de conjuntos recursivos enumeráveis sobre A por analogia a noções dos casos discretos. Um conjunto $S \subseteq A^l$ é **recursivo enumerável** sobre A se existe uma máquina BSS M com espaço de entrada A^l, tal que $\Omega_M = S$. S é **decidível ou recursivo** sobre A se ambos S e S^c (o complementar de S) são recursivos enumeráveis sobre A.

Exemplo 1 *Uma máquina BSS que computa o maior inteiro em x, $\lfloor x \rfloor$, para $x \geq 0$ em $\mathbb{R}$ pode ser vista na figura 1:*

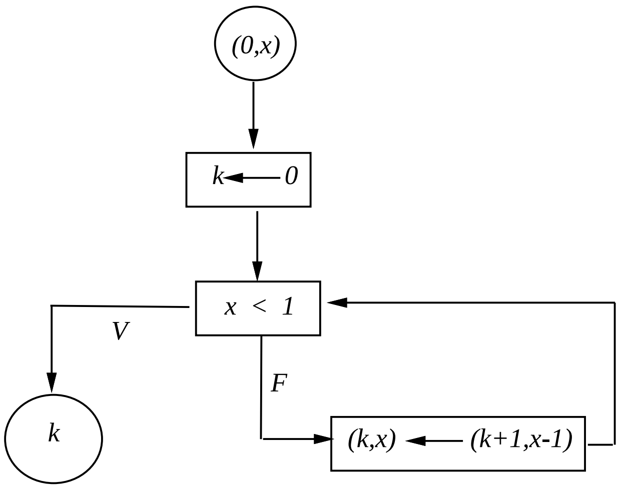

Figura 1: Diagrama de fluxo de uma máquina BSS que computa o maior inteiro em x

Observe, na figura 1, que para $x > 0$ em R, o custo da computação $\lfloor x \rfloor$ é de $\lfloor x+1 \rfloor$ comparações e $\lfloor x \rfloor$ de aritmética básica computacional. Usando busca binária este custo pode ser reduzido para $O(log(x+1))$ mas essencialmente, não mais em nosso modelo. Ele é interessante para observar que em modelos de computação onde $\lfloor x \rfloor$ tão bem quanto as operações aritmética *básica podem ser computadas em tempo constante, aparentemente problemas difíceis tal qual fatoração de inteiros e testes de satisfiabilidade de fórmulas proposicionais podem ser resolvidas eficientemente em tempo polinomial.*

4 Introdução à matemática intervalar

Um *Intervalo Real* $[a, b]$ foi definido como um subconjunto de $\mathbb{R}$ dado por: $[a, b] = x \in \mathbb{R}|a \leq x \geq b$, e o conjunto de todos os intervalos reais é denotado por $I(\mathbb{R})$. Observe que a finalidade principal em se considerar um intervalo seria o fato deste intervalo constituir uma aproximação dos reais que ele contém. As operações aritméticas com intervalos são as definidas a seguir:

i. $[a, b] + [c, d] = [a + c, b + d]$

ii. $[a, b] - [c, d] = [a - d, b - c]$

iii. $[a, b] \cdot [c, d] = [min\{a.c, b.c, a.d, b.d\}, max\{a.c, b.c, a.d, b.d\}]$

iv. $[a, b]/[c, d] = [a, b] \times [1/d, 1/c]$, se $0 \notin [c, d]$

Em [Moo66] e [Moo79] pode-se observar que o conjunto $I(\mathbb{R})$ constitui uma estrutura algébrica que generaliza a estrutura dos reais. A aritmética de intervalos satisfaz a propriedade da inclusão monotônica, isto é, se $I, J, K, L \in I(\mathbb{R})$ e $I \subseteq K$ e $J \subseteq L$, então:

i. $I + J \subseteq K + L$

ii. $I - J \subseteq K - L$

iii. $I \cdot J \subseteq K \cdot L$

iv. $I/J \subseteq K/L$ (Se $0 \notin J, L$).

A imagem de uma função real $f : D \subseteq \mathbb{R} \to \mathbb{R}$ sobre um conjunto $X \subseteq D$ é $f'(X) = \{f(x)|x \in X\}$. Seja o intervalo real fechado e limitado $X = [a, b]$. Se f é contínua, então $f'(X)$ também é um intervalo da mesma espécie. Um problema fundamental típico na análise intervalar é o cálculo de $f'(X)$ ou no mínimo uma boa aproximação dele. Se f é definida em termos de operações aritméticas e funções com extensões intervalares, então o uso de computação intervalar resulta em uma função intervalar F tal que: $f(X) \subseteq F(X)$, para $X \subseteq D$ [Moo66]. Este cálculo tem a vantagem de ser completamente automático e de não requerer conhecimento das propriedades específicas de f.

Como a análise intervalar é uma teoria matemática cujo principal objetivo é responder a questões de exatidão e eficiência que surgem na prática da computação científica, é razoável se esperar que técnicas intervalares forneçam a garantia do controle de erros e possam ser aplicadas quase que automaticamente. Considerando técnicas intervalares o diâmetro de um intervalo solução é um indicativo da influência do erro no resultado final. Entretanto, respostas intervalares não garantem que estejam incluídos algo do nosso interesse. Para isso é preciso uma fundamentação matemática acurada em cada estágio do projeto e implementação do algoritmo. Para tirarmos partido das vantagens das técnicas intervalares, os algoritmos devem ser intervalares, isto é, eles não devem ser uma versão intervalar de um algoritmo pontual.

5 A máquina BSS-Intervalar

A definição de máquinas BSS-Intervalar é análoga à definição do modelo BSS clássico, só que nesta nova máquina substituímos a condição de se trabalhar sobre anéis ordenados por quasi-anéis. Esta alteração não trazerá mudanças no resto da definição da máquina, pois as propriedades imprescindíveis de anéis para máquinas BSS estão também presente em quasi-anéis. Como pela proposição 1 todo anel é um quasi-anel, e portanto todo anel ordenado é um quasi-anel ordenado, podemos concluir que todas as máquinas BSS clássicas também são máquinas do novo modelo. Assim, tudo o que pode ser computado por máquinas BSS também podem ser computados por máquinas BSS-Intervalar. Além disso, existem estruturas que são quasi-anéis mas não são anéis, por exemplo os intervalos reais munidas das operações de adição + e produto $\cdot$. Portanto máquinas BSS-Intervalar fazem computações que não são possíveis de serem efetuadas usando máquinas BSS clássicas.

É necessário fazermos uma análise quanto a estrutura algébrica que o conjunto dos intervalos reais formam. Se estes formassem um anel, uma máquina BSS poderia trabalhar sob ele, porém por não satisfaz as condições de distributividade e associatividade dos anéis definiremos a seguir uma estrutura mais abrangente que os anéis, a qual chamaremos de quasi-anéis que formará um conjunto onde os intervalos reais fazem parte.

Definição 3 *Seja R um conjunto não vazio sobre o qual estão definidas duas operações binárias $+$ e $\cdot$, denominadas, respectivamente, adição e multiplicação . Estas estruturas são* **quasi-anéis** *se para quaisquer $a, b, c \in R$. $(R, +, \cdot)$ as seguintes propriedades sejam satisfeitas:*

i. $a + b = b + a \Rightarrow$ *lei da comutatividade da adição*

ii. Existe $\mathbb{O} \in R$ *tal que* $a + \mathbb{O} = \mathbb{O} + a = a$, $\forall a \in R \Rightarrow$ *existência de um zero (elemento neutro)*

iii. Existe $1 \in R$ *tal que* $1a = a1 = a$, $\forall a \in R \Rightarrow$ *existência de uma unidade*

iv. Existe um elemento $k \in R$ *tal que:*

iv_1. $k + 1 = \mathbb{O}$

iv_2. $k(ab) = (ka)b = a(kb)$

iv_3. $k(a + b) = (ka) + (kb)$

Um quasi-anel A é dito **ordenado** se A possui um subconjunto P que admite o seguinte:

1. se $a, b \in P$ então $a + b$, $a \cdot b \in P$

2. Seja $\mathbb{O}$ o elemento neutro e $-a$ o inverso de $(A, +)$. Para todo $a \in A, a/\mathbb{O}$ e $a \in P$ ou $-a \in P$, nunca ambos.

Proposição 6 *todo anel é um quasi-anel [Aar99].*

Provaremos adiante que, os intervalos reais satisfazem as condições de quasi-anel ordenado. Definiremos com isso, uma máquina BSS estendida para trabalhar sobre quasi-anéis ordenados e sendo o conjunto dos intervalos reais um quasi-anel ordenado, essa máquina BSS será capaz de trabalhar sobre essa estrutura.

Sendo assim, faz-se necessário a seguinte proposição :

Proposição 7 $[I(\mathbb{R}), +, \cdot]$ *é um quasi-anel ordenado [Aar99]*

Exemplo 2 *A máquina BSS intervalar newtoniana*

O método de Newton real é um algoritmo que serve para calcular a raiz de uma dada equação, através da construção de uma seqüência convergente de pontos da reta real. De maneira análoga, a versão intervalar do método de Newton permite contruir uma seqüência convergente de intervalos, cujo limite será um intervalo que contém a raiz real da função dada.

Por ser um método autovalidável, se tomarmos um intervalo inicial que não contenha a raiz real, então, numa dada iteração, iremos obter um intervalo vazio como resultado. Caso contrário, se tomarmos um intervalo inicial, que contém a raiz real da equação $f(x) = 0$, e, considerando que a seqüência intervalar que se obtém é de intervalos encaixados, então obteremos como limite o intervalo de menor diâmetro possível, que ainda contém a raiz real desejada. Na prática, esta é a vantagem do uso do método de Newton intervalar.

Veremos agora, como se pode definir uma versão intervalar para o método de Newton real:

A primeira idéia que se tem para definir o método de Newton intervalar é tomar uma extensão intervalar para o operador Newtoniano real, ou seja, definir $N(X) = X - \frac{F(X)}{F'(X)}$, onde $F(X)$ e $F'(X)$ são extensões intervalares para as funções $f(x)$ e $f'(x)$ (função real contínua e derivada contínua, respectivamente).

Agora, veremos porque o método de Newton construído dessa maneira é sempre divergente.

Por exemplo, para calcularmos a raiz $\sqrt{2}$ da função $f(x) = x^2 - 2 = 0$ no intervalo $[1, 2]$; temos:

$$F(X) = X^2 - [2, 2] \text{ e } F'(X) = [2, 2] \cdot X.$$

Tomando $X_0 = [1, 2]$, calculamos

$$\begin{aligned} X_1 &= X_0 - \frac{F(X_0)}{F'(X_0)} \\ &= [1, 2] - \frac{[1,2]^2 - [2,2]}{[2,2] \cdot [1,2]} \\ &= [1, 2] - \frac{[1,4] - [2,2]}{[2,4]} \\ &= [1, 2] - \frac{[-1,2]}{2,4} \\ &= [1, 2] - [-1, 2] \cdot [\frac{1}{4}, \frac{1}{2}] \\ &= [1, 2] - [-\frac{-1}{2}, 1] \\ &= [1, 2] + [-1, \frac{1}{2}] \\ &= [0, \frac{5}{2}] \end{aligned}$$

Observe que:

$$diam(X_1) = diam([0, \tfrac{5}{2}]) = \tfrac{5}{2} > 1 = diam(X_0);$$

e que $X_0 \subseteq X_1$. Observe também que:

$$F'(X_1) = [2,2] \cdot X_1 = [2,2] \cdot [0, \tfrac{5}{2}] = [0,5]$$

contém o zero, logo, na próxima interação (cálculo de X_2), não poderemos efetuar a operação de divisão.

Como queremos que a seqüência de intervalos X_k convirja para o intervalo pontual $[x_*, x_*]$ (onde x_* é a raiz), é necessário que $diam(X_k) \to 0$, quando $k \to +\infty$. Mas como,

$$diam(X_{k+1}) = diam(X_k - \tfrac{F(X_k)}{F'(X_k)}) = diam(X_k) + diam(\tfrac{F(X_k)}{F'(X_k)}) > diam(X_k),$$

conclui-se que a seqüência de intervalos definida por $X_{k+1} = X_k - \frac{F(X_k)}{F'(X_k)}$ é sempre divergente.

Assim, devemos modificar o método de maneira que o diâmetro de um intervalo recém calculado X_{k+1} diminua em relação ao diâmetro de X_k e que ainda contenha x_*. A idéia agora é construirmos uma seqüência de intervalos encaixados $X_0 \supseteq X_1 \supseteq X_2 \supseteq X_3 \supseteq X_4 \supseteq \cdots \supseteq X_k \supseteq \cdots \supseteq [x_*, x_*]$, cujos diâmetros vão diminuindo à medida que o valor de k aumenta. Para tanto, vamos avaliar com intervalos somente a derivada, cuidando para que o intervalo reultante da avaliação intervalar da derivada não contenha o zero.

Desta forma, precisamos exigir que a função f satisfação certas condições no intervalo inicial X_0, ou seja, para que a avaliação intervalar da derivada não contenha o zero, é necessário que a função f não tenha pontos críticos (máximos ou mínimos locais) em X_0. Veremos, a seguir, como definir o método de Newton intervalar que sempre é convergente.

Teorema 1 *Seja $f(x)$ uma função real contínua em $X_0 = [a,b]$, de modo que $f(a) \cdot f(b) < 0$. Assim, existe $x_* \in X_0$ tal que $f(x_*) = 0$. Calculamos um intervalo $M = [m_1, m_2]$, tal que $0 < m_1 \leq \frac{f(x)-f(y)}{x-y} \leq m_2 < +\infty$, para todo $x, y \in X_0$, com $x \neq y$, que servirá como uma inclusão intervalar para a imagem intervalar da derivada $f'(x)$ da função $f(x)$.*

Finalmente definindo o **operador Newtoniano intervalar** *por:*

$$N(X) = med(X) - \tfrac{f(med(X))}{M},$$

constuímos a seqüência intervalar recursiva:

$$X_{k+1} = X_k \cap N(X_k) \text{ e } X_0 = [a,b], \text{ que tem as seguintes propriedades:}$$

1. $x_* \in X_k, \forall k \geq 0$

2. $X_0 \supseteq X_1 \supseteq X_2 \supseteq X_3 \supseteq \cdots$ *e* $lim_{k \to +\infty} X_k = [x_*, x_*]$

3. $diam(X_{k+1}) \leq (1 - \frac{m_1}{m_2}) \cdot diam(X_k)$, *ou seja,* $diam(X_{k+1}) \leq (1 - \frac{m_1}{m_2})^k \cdot diam(X_0)$ *o que implica que* $lim_{k \to \infty} diam(X_k) = 0$

4. *Se existe $k_0 \geq O$ tal que $X_{k_0} \cap N(X_{k_0}) = \emptyset$, então não existe raiz da função f no intervalo inicial X_0.*

A última condição, garante a autovalidação do algoritmo, ou seja, se escolhermos um intervalo inicial que não contém raiz da função, então o método pára, caso contrário, ele calcula o intervalo limite da seqüência X_k, que contém x_*, que é a raiz da função f, validando a resposta.

Observações

1. $x_* \in x - \frac{f(x)}{M}, \forall x \in X_0$.
2. *Podemos tomar* $M_1 = Im(\frac{f(x)}{x-x_*}, X_0) \subseteq M$
   ou ainda $M_2 = [m_1, m_2] = Im(f'(x), X_0) \subseteq M$.

Calcular a raiz da função $f(x) = x^2 - 2 = 0$ *em* $X_0 = [1, 2]$

$f(x) = x^2 - 2$ *é uma função contínua em* $\mathbb{R}$, *logo é contınua no intervalo inicial* $X_0 = [1, 2]$. *Temos que:*

$$f(1).f(2) = (1^2 - 2) \cdot (2^2 - 2) = (-1) \cdot (2) = -2 < 0,$$

logo a função tem pelo menos uma raiz no intervalo $[1, 2]$.

O cálculo da inclusão intervalar M *para a derivada pode ser feito por:*

$$\begin{aligned} \frac{f(x)-f(y)}{x-y} &= \frac{(x^2-2)-(y^2-2)}{x-y} \\ &= \frac{x^2-y_2}{x-y} \\ &= \frac{(x-y)\cdot(x+y)}{x-y} \\ &= x + y \end{aligned}$$

e daí, temos que:

$$0 < 2 \leq \frac{f(x)-f(y)}{x-y} = x + y \leq 4,$$

para quaisquer $x, y \in X_0 = [1, 2]$, *pois* $1 \leq 2$ *e* $1 \leq y \leq 2$ *logo,* $1 + 1 \leq x + y \leq 2 + 2$.

Como o cálculo da derivada $f'(x)$ *da função* $f(x)$ *é, este caso, mais fácil, podemos tomar:*

$$\begin{aligned} M &= [m_1, m_2] \\ &= Im(f'(x), X_0) \\ &= Im(2 \cdot x, [1, 2]) \\ &= [min\{2 \cdot x/x \in [1, 2]\}, max\{2 \cdot x/x \in [1, 2]\}] \\ &= [2, 4] \end{aligned}$$

Assim, percebemos que as duas inclusões que foram calculadas para a derivada intervalar coincidem.

O operador intervalar Newtoniano fica, então:

$$N(X) = [med(X), med(X)] - \frac{[(med(X))^2-2,(med(X))^2-2]}{[2,4]}$$

Finalmente, construímos a seqüência intervalar recursiva

$$X_{k+1} = X_k \cap N(X_k) \ e \ X_0 = [1, 2]$$

Computando os valores da seqüência, temos que o valor da raiz da função $f(x) = x^2 - 2$ *é um valor real* $x_* \in [1.414213562373094, 1.414213562373096]$

Vejamos, na figura 2, um modelo BSS intervalar que computa a raiz de uma função pelo método de Newton.

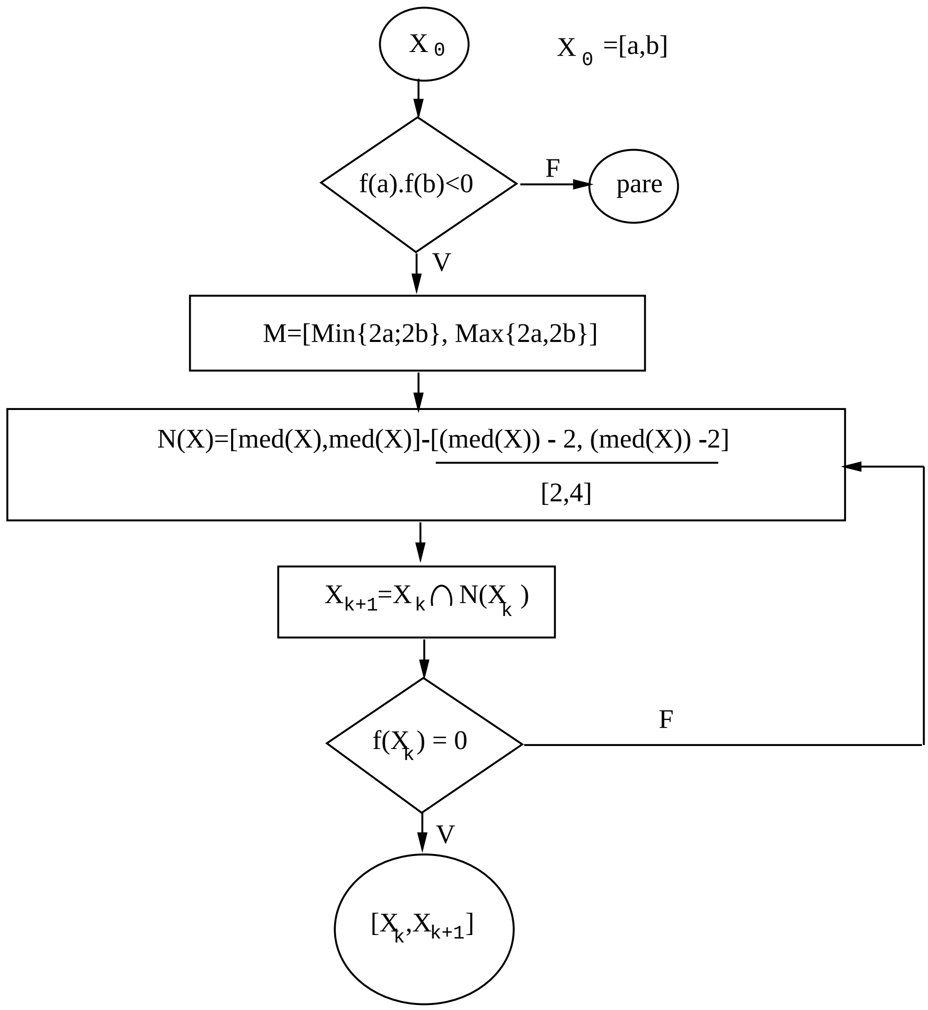

Figura 2: Máquina BSS intervalar que calcula as raízes da função $f(x) = x^2 - 2$

6 Computabilidade sobre o Espac co de Intervalos

Seja $\mathbb{I}(\mathbb{R})$ o conjunto de todos os intervalos com extremos reais ou, simplesmente, intervalos reais.

Definimos sobre $\mathbb{I}(\mathbb{R})$ a ordem $\sqsubseteq_{\Re}$ por

$$[r_1, s_1] \sqsubseteq_{\Re} [r_2, s_2] \text{ se e somente se } r_1 \leq r_2 \text{ e } s_2 \leq s_1,$$

$\leq$ é a relação usual de ordem em $\mathbb{R}$.

Proposição 8 (Sco70) *O conjunto parcialmente ordenado* $\Re = \langle \mathbb{I}(\mathbb{R}), \sqsubseteq_{\Re} \rangle$ *é um dcpo.*

Uma cadeia de intervalos reais $(I_i)_{i \in \mathbb{N}}$ *converge* para um intervalos real I, se $I_i \sqsubseteq_{\mathbb{I}(\mathbb{R})} I$ para cada $i \in \mathbb{N}$ e para algum outro $J \in \mathbb{I}(\mathbb{R})$ com a propriedade $I \sqsubseteq_{\Re} J$. Isto é, se $I = \bigsqcup(I_i)_{i \in \mathbb{N}}$.

Proposição 9 *Seja* $([a_i, b_i])_{i \in \mathbb{N}}$ *uma cadeia de intervalos reais (racionais). Onde,* $([a_i, b_i])_{i \in \mathbb{N}}$ *converge para os intervalos reais.*

$$[\lim_{i \to \infty} a_i, \lim_{i \to \infty} b_i].$$

Demonstração: Se $([a_i, b_i])_{i \in \mathbb{N}}$ é uma cadeia de intervalos reais (racionais) então ela converge para algum intervalo real $[r, s]$. Por este motivo, $a_i \leq r \leq s \leq b_i$ para cada $i \in \mathbb{N}$ e para cada intervalo $[t, u]$ tal que $a_i \leq t \leq u \leq b_i$ para cada $i \in \mathbb{N}$, $r \leq t \leq u \leq s$. Desta forma, $\lim_{i \to \infty} a_i = r$ e $\lim_{i \to \infty} b_i = s$. Assim, $[r, s] = [\lim_{i \to \infty} a_i, \lim_{i \to \infty} b_i]$. ■

A garantia de convergência permiti-nos simplificar, w.r.t. a noção similar de computabilidade real, o conceito de computabilidade intervalar.

Definição 4 *Um intervalo real I é* **computável** *se existir uma cadeia enumerável recursiva de intervalos racionais convergentes para I.*

Os *Intervalos Computáveis* são os intervalos reais os quais são supremos de cadeias r.e. de intervalos reais.

Proposição 10 *Seja x e y números reais tal que $x \leq y$. $[x, y]$ é computável se e somentew se x e y são Cauchy-computàvel.*

Demonstração: Se $[x, y]$ é computável então $\{[p, q] \in \mathbb{I}(\mathbb{Q}) : \ p < x \leq y < q\}$ é r.e. e, então os conjuntos $\{p \in \mathbb{Q} : \ p < x\}$ e $\{q \in \mathbb{Q} : \ y < q\}$ são r.e. Assim, podemos obter uma sequência r.e $p_0, p_1, \cdots, p_n, \cdots$ de números racionais tal que p_0 é uma parte inteira de x, $p_0 < p_1 < \cdots < p_n < \cdots$ e $x = \lim_{n \to \infty} p_n$. Assim, definimos $p'_0 = p_0 + 1$ e $p'_{i+1} = 2p_{i+1} - p_i = p_{i+1} + p_{i+1} - p_i$ obtendo uma sequência r.e. tal que $p'_0 > p'_1 > \cdots > p'_n > \cdots$

$$\begin{aligned}
\lim\nolimits_{n \to \infty} p'_n &= \lim\nolimits_{n \to \infty} p'_{n+1} \\
&= \lim\nolimits_{n \to \infty} 2p_{n+1} - p_n \\
&= 2 \lim\nolimits_{n \to \infty} p_{n+1} - \lim\nolimits_{n \to \infty} p_n \\
&= 2x - x \\
&= x
\end{aligned}$$

Dessa maneira, o número real x é Cauchy-computável. Da mesma forma, o número real y é Cauchy-computável.

Por outro lado se x e y são Cauchy-computáveis então por definição de computabilidade dos números reais, estudada por Martin-Löf em [Lof70], os conjuntos $\twoheaddownarrow x = \{[p, q] \in \mathbb{I}(\mathbb{Q}) : \ p < x < q\}$

e $\Downarrow y = \{[p,q] \in \mathbb{I}(\mathbb{Q}) : p < y < q\}$ são r.e. Assim $\Downarrow x \cap \Downarrow y$ da mesma maneira é r.e. Desta forma, porque $\Downarrow x \cap \Downarrow y = \{[p,q] \in \mathbb{I}(\mathbb{Q}) : p < x \leq y < q\}$ o intervalo real $[x,y]$ é computável. ■

Um intervalos real $[a,b]$ é computável se, e somente se, ambos extremos são números reais Cauchy-computáveis. Dessa forma, a computabilidade em intervalos reais degenerados[4].

Proposição 11 *Seja $\mathbb{I}(\mathbb{R})_C$ o conjunto de todos os intervalos reais computáveiws. Então*

1. *$\mathbb{I}(\mathbb{R})_C$ é um conjunto contável e, dessa maneira, $\mathbb{R}_C \neq \mathbb{R}$.*
2. *$\mathbb{I}(\mathbb{R})_C$ não é um conjunto r.e.*
3. *Não é decidível, então dois intervalos reais arbitrários são iguais ou não.*

Seja $\langle \mathbb{I}(\mathbb{Q}), \sqsubseteq_{\mathbb{I}(\mathbb{Q})} \rangle$ um conjunto parcialmente ordenado, onde $\mathbb{I}(\mathbb{Q})$ é o conjunto de intervalos racionais e $\sqsubseteq_{\mathbb{I}(\mathbb{Q})}$ é a inclusão de ordem reversa sobre $\mathbb{I}(\mathbb{Q})$, isto é, $[a,b] \sqsubseteq_{\mathbb{I}(\mathbb{Q})} [c,d]$ se e somente se $a \leq c$ e $d \leq b$. Cada função monotônica $g : \mathbb{I}(\mathbb{Q}) \longrightarrow \mathbb{I}(\mathbb{Q})$ preserva as cadeias, isto é, se $(I_i)_{i \in \mathbb{N}}$ é a cadeia em $\mathbb{I}(\mathbb{Q})$ então $(g(I_i))_{i \in \mathbb{N}}$ também é uma cadeia em $\mathbb{I}(\mathbb{Q})$.

Definição 5 *Uma função $f : \mathbb{I}(\mathbb{R}) \longrightarrow \mathbb{I}(\mathbb{R})$ é* **computável** *se existe uma função monotônica e Church-Computável $g : \mathbb{I}(\mathbb{Q}) \longrightarrow \mathbb{I}(\mathbb{Q})$ tal que para cada intervalo real I e cada cadeia de intervalos racionais $(I_i)_{i \in \mathbb{N}}$ convergentes para I*

$$f(I) = \lim_{i \in \mathbb{N}} g(I_i)$$

Nossa noção de computabilidade sobre os Intervalos Reais, $\mathbb{I}(\mathbb{R})$, é baseado na computabilidade sobre os Intervalos Racionais $\mathbb{I}(\mathbb{Q})$, é a clássica computabilidade Church-Turing. Esta noção de computabilidade implica em ordem-continuidade, Uma função g satisfazendo a definição de computabilidade, é a própria $f_{/\mathbb{I}(\mathbb{Q})}$, a função f restrita para o conjunto $\mathbb{I}(\mathbb{Q})$.

Teorema 2 *Se $f : \mathbb{I}(\mathbb{R}) \longrightarrow \mathbb{I}(\mathbb{R})$ é uma função computável então f é uma função de ordem-contínua w.r.t. $\Re$.*

DEMONSTRAÇÃO: Devemos provar que f é monotônica. Seja $[r_1,s_1],[r_2,s_2] \in \mathbb{I}(\mathbb{R})$ tal que $[r_1,s_1] \sqsubseteq_{\Re} [r_2,s_2]$. Se $(I_i)_{i \in \mathbb{N}}$ e $(J_i)_{i \in \mathbb{N}}$ são cadeias de intervalos racionais convergindo para $[r_1,s_1]$ e $[r_2,s_2]$, respectivamente, então, pela monotonicidade, $g((I_i)_{i \in \mathbb{N}})$ e $g((J_i)_{i \in \mathbb{N}})$ são cadeias. Mas, pela computabilidade de f, $f([r_1,s_1]) = \lim_{i \in \mathbb{N}} g(I_i)$ e $f([r_2,s_2]) = \lim_{i \in \mathbb{N}} g(J_i)$. Assim, considerando a noção de limite sobre os intervalos reais, $\lim_{i \in \mathbb{N}} g(I_i) \sqsubseteq_{\Re} \lim_{i \in \mathbb{N}} g(J_i)$. Desta forma, $f([r_1,s_1]) \sqsubseteq_{\Re} f([r_2,s_2])$ e assim f é monotônica. Por outro lado, se $(I_i)_{i \in \mathbb{N}}$ é uma cadeia de intervalos racionais convergindo para $[r,s]$.

$$\begin{aligned} f(\bigsqcup(I_i)_{i \in \mathbb{N}}) &= f([r,s]) \\ &= \lim_{i \in \mathbb{N}} g(I_i) \\ &= \bigsqcup\{g(I_i) \mid i \in \mathbb{N}\} \\ &= \bigsqcup\{f_{/\mathbb{I}(\mathbb{Q})}(I_i) \mid i \in \mathbb{N}\} \\ &= \bigsqcup\{f(I_i) \mid i \in \mathbb{N}\} \end{aligned}$$

[4] Um intervalos degenerado I é um intervalo onde os seus extremos são iguais, ou seja, um intervalo do tipo r,r onde r é um número Cauchy-computável

Assim, a função f também é contínua. ■

Claramente, nós podemos estender, naturalmente, este conceito para $\mathbb{I}(\mathbb{R})^n$. A seguir enumeramos algumas funções intervalares computáveis.

Proposição 12 *As seguintes funções de $\mathbb{I}(\mathbb{R})^n$ para $\mathbb{I}(\mathbb{R})$ são computáveis:*

1. *As funções aritiméticas e as funções identidade são computáveis*[5]
2. $$sen[r,s] = \begin{cases} [-1,1] & , \textit{if } 2\pi \le s-r \\ [minS, 1] & , \textit{if } \pi \le s-r < 2\pi \textit{ and } sen\ r + \frac{s-r}{2} > 0 \\ [-1, maxS] & , \textit{if } \pi \le s-r < 2\pi \textit{ and } sen\ r + \frac{s-r}{2} < 0 \\ [minS, maxS] & , \textit{outro caso} \end{cases}$$

   onde $S = \{sen\ r, sen\ s\}$.
3. $cos\ [r,s] = sen\ [r + \frac{\pi}{2}, s + \frac{\pi}{2}]$, $tg\ I = \frac{sen\ I}{cos\ I}$ *and* $sec\ I = \frac{1}{cos\ I}$
4. *Cada função intervalar polinomial* $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$, *onde o* a_i*'s são intervalos computáveis.*

DEMONSTRAÇÃO: A primeia destas funções , chamada de aritmética intervalar, pode ser recuperada a partir de restrições de intervalos racionais, que são, claramente, computáveis. A função constante para um intervalo real computável c pode ser recuperado a partir de uma função monotônica computável Church-Turing.

$$g(I) = \begin{cases} I & , \text{if } c \in I \\ [-\infty, +\infty] & , \text{em outro caso} \end{cases}$$

Aqui, usamos um intervalo especial, $[-\infty, +\infty]$ para denotar o "intervalo" de toda a reta real, isto é, um "intervalo" é para suprir a necessidade de tratar com funções parciais.

As funções de valor absoluto, exponencial, logaritmica e seno são recuperadas a partir de restrições de intervalos racionais que são trivialmente Church-Turing computáveis. As funções restantes são baseadas sobre o seno, operações aritméticas e as funções constantes π, que são todas computáveis. Finalmente, porque a soma, produto e os coeficientes são computáveis, a função intervalar polinomial com coeficientes computvéis podem ser recuperadas a partir da função $q : \mathbb{I}(\mathbb{Q}) \longrightarrow \mathbb{I}(\mathbb{Q})$ definida por $q(I_i) = a_{0,i} + a_{1,i}I + \cdots + a_{n,i}I^n$, onde $a_{k,i}$ é o intervalo racional i-ésimo na enumeração efetiva de a_k. Assim, porque q é claramente Church-Turing computável, p é computável. ■

Um exemplo de função não computável é a igualdade com zero, isto é, a função $ig) : \mathbb{I}(\mathbb{R}) \longrightarrow \mathbb{I}(\mathbb{R})$ definida por

$$ig0([r,s]) = \begin{cases} [1,1] & , \text{if } [r,s] = [0,0] \\ [0,0] & , \text{em outro caso} \end{cases}$$

[5] visto que os logaritmos reais são funções de $\mathbb{R}^+$ (conjunto de números positivos reais) dentro de $\mathbb{R}$, esta versão intervalar é uma função de $\mathbb{I}(\mathbb{R})^+$ para $\mathbb{I}(\mathbb{R})$.

Isto acontece porque, trivialmente esta função não é monotônica ($[-1,1] \sqsubseteq [0,0]$. Mas $ig0([-1,1]) = [0,0] \not\sqsubseteq ig0([0,0]) = [1,1]$) e assim não contínua. Assim, ig) não é uma função computável. Também podemos pensar em $\{0,1\}$, isto é, $\{[0,0],[1,1]\}$ um conjunto com dois elementos, mas independentemente de como os ordenamos, estas duas funções resta não ser contínua.

A computabilidade intervalar preserva a composição e o produto cartesiano, isto é se, $f : \mathbb{I}(\mathbb{R})^n \longrightarrow \mathbb{I}(\mathbb{R})^m$, $g : \mathbb{I}(\mathbb{R})^m \longrightarrow \mathbb{I}(\mathbb{R})^p$, $h : \mathbb{I}(\mathbb{R})^r \longrightarrow \mathbb{I}(\mathbb{R})^s$ são funções computáveis assim $g \circ f : \mathbb{I}(\mathbb{R})^n \longrightarrow \mathbb{I}(\mathbb{R})^p$ e $f \times h : \mathbb{I}(\mathbb{R})^{n+r} \longrightarrow \mathbb{I}(\mathbb{R})^{m+s}$ definidas por $g \circ f(I) = g(f(I))$ e $f \times h(I,J) = (f(I), h(J))$ também são funções computáveis.

7 Resultados Finais

A estrutura algébrica que suporta os algorítmos em computação científica é apenas um anelóide ou um quasi-anel (uma estrutura parecida com um anel) [Aar99]. Essa estrutura deve, também, ser considerada quando se deseja realizar uma fundamentação para a computação científica.

Nosso artigo adaptou este modelo para intervalos, para isto generalizamos os conceitos básicos das máquinas BSS. Por exemplo, como o espaço dos intervalos não é um anel e sim um quasi-anel analisamos quais as implicações de se trabalhar com esta estrutura. Tivemos certos cuidados, como por exemplo, garantir que quando a estrutura for um anel esta máquina se comporte da mesma maneira que BSS.

Uma questão fundamental que analisamos neste trabalho foram as conseqüências de se atribuir uma estrutura de quasi-anel em substituição a um anel conforme propuseram BSS em sua máquina. Esta modificação não causou nenhum efeito, pois a única parte de BSS que usa propriedades de anel são os nós computacionais, mas, as propriedades que são usadas são também propriedades de quasi-anéis.

Por outro lado provamos que os intervalos reais são quasi-anéis ordenados, que sendo a máquina BSS capaz de manipular com quasi-anéis também será capaz de trabalhar sobre os intervalos reais, já que estes estão contidos nesta estrutura.

Além disso, nós introduzimos a noção de computabilidade sobre os intervalos reais baseado na computabilidade sobre os intervalos racionais, isto é computabilidade clássica. Assim, a função computa aproximações finitas de um objeto finito no caminho do limite, ele computa o objeto ideal.

A computabilidade intervalar aqui definida tem diversas propriedades que a teoria da computabilidade sobre intervalos de análises considera. Esta noção de computabilidade é baseada sobre uma noção similar usando a consolidada teoria dos domínios, o que nos traz toda essa confiabilidade desta teoria para a construção de uma análises computável.

A vantagem da nossa definição é que ela é interna, no sentido que não é baseada na computabilidade dos números reais, resultando na nolção de computabilidade sobre os intervalos reais, mais intuitivo e simples.

Finalmente, podemos definir uma estensão natural da máquina de Turing baseada na abordagem da computação sobre os intervalos reais baseado nas computações , em um sentido clássico, sobre os intervalos racionais. Uma importante contribuição da máquina **C**-Turing é que podemos mostrar funções intervalares reais computadas em vez de apenas funções intervalares reais computadas.

Assim, neste trabalho apresentamos duas abordagens de natureza diferente como uma alternativa para o estudo de uma análise intervalar computável.

Resumo

Existem diferentes abordagens para a computabilidade nos números reais, e, uma diferenc ca sobre estas abordagens é o caminho em que o número real é representado. Basicamente, existem duas linhas de pesquisa na computabilidade dos contínuos. Na primeira estimativa, uma saída arbitrária é computada a partir de uma entrada de aproximação Bra95]. A segunda abordagem para a computabilidade dos reais foi desenvolvida por Blum, Shub e Smale [BSS89]. Em que, nas máquinas BSS, um número real é visto como uma entidade acabadas e a funções computáveis geradas a partir de uma classe de funções básicas (no similar caminho para as funções parciais recursivas).

Neste Trabalho, estudamos este modelo BSS, e estendemos para a computabilidade no espac co dos intervalos reais.

key-words: Computabilidade, Teoria da Computação , Intervalos Reais, Modelo BSS and Espac co Contínuo.

Abstract

Different approaches exist for the computability in the real numbers, and, an important difference among these approaches is the way the real number is represented. Basically, there are two lines of research in continuous computability. In the first estimation of, an output arbitrary accuracy is computed from an input aproximation [Bra95]. The second approach to real computability was developed by Blum, Shub and Smale [BSS89]. Where, the BSS machines, a real number is seen as a finished entity and the computable functions are generated starting from a class of basic functions (in a similar way to the partial recursive functions). In this work, we study this BSS model, used to characterize a computability theory on the real numbers and we extend it to model computability in the real intervals space.

key-words: Computability, Computation Theory, Real Intervals, BSS Model and Continuous Spaces.

Referências

[Aar99] Aarão Lyra. *Computabilidade nos Espaços dos Intervalos Reais: Um Modelo BSS Intervalar*. Dissertação de Mestrado. DIMAp da UFRN, Natal/RN, 1999.

[Aci91] Benedito Melo Acióly. *Computational Foundation of Interval Mathematics* (in portuguese). PhD thesis, CPGCC da UFRGS, Porto Alegre, 1991.

[BA97a] Benjamin René Callejas Bedregal and Benedito Melo Acióly. *Moore-Computability: A Proposal for a Computable Interval Analysis*. Revista de Informática Teórica e Aplicada, Porto Alegre-RS, 1997.

[BA97b] Benjamin René Callejas Bedregal and Benedito Melo Acióly. *Computability on the Interval Space: A domain approach*. XXIII Conferência Latinoamericana de Informática-CLEI, Valparaí.so-Chile, Novembro de 1997.

[Bed96] Benjamín R. Callejas Bedregal. *Continuous Information Systems: A Computational and Logical Approach to Interval Mathematics* (in portuguese). PhD thesis, UFPE-Depto. de Informática, Recife, 1996.

[BL74] Walter S. Brainerd and Lawrence H. Landweber. *Theory of Computation*. John Wiley & Sons, U.S.A., 1974.

[Bol95] Paolo Boldi. *Computability and complexity over the reals*. University of Milan, Italy, February, 1995.

[Bra95] Vasco Brattka. *Recursive Characterization of Computable Real-valued Functions and Relations*. Preprint submitted to Elsiever Science, 1995.

[BSS89] L. Blum, M. Shub and S. Smale. *On a Theory of Computation and Complexity over Real Number: NP-completness, recursive functions and universal machines*. Bull. of the Amer. Math. Soc. 21, 1989, pg. 1-46.

[BSSC95] L. Blum, M. Shub, S. Smale and Felipe Cucker. *Complexity and Real Computation: A Manifesto*. 1995.

[Dim97] Graçaliz Pereira Dimuro. *Uma Representação Computacional Global para Sistemas Ordenados de 2. Ordem em Espaços Coerentes Intervalares Bi-Estruturados, com Aplicação em Matemática intervalar*, Tese de Doutoramento, Porto Alegre-RS, Brasil, Junho de 1997.

[Gia93] Pietro Di Gianantonio. *A Functional Approach to Computability on Real Numbers*. PhD thesis, Università di Pisa-Genova-Udine, Italy, march 1993.

[Grz57] A. Grzegorczyk. *On the definitions of computable real continuous functions*. Fundamenta Mathematicae, 44(1), 1957.

[ML70] Per Martin-Löf. *Notes on Contructive Mathematics*. Almqvist & Wiksell, Stockholm, 1970.

[Moo66] R. E. Moore. *Interval Analysis.* Englewoods Cliffs: Prentice Hall, 1966.

[Sco70] Dana S. Scott. Outline of a mathematical theory of computation. In *4 th. annu Princeton Conference on Inf. Science and Systems*, pages 65-106, 1970.