



UFC – Universidade Federal do Ceará – Campus Quixadá
Av. José de Freitas Queiroz, 5003, Quixadá - CE, 63902-580
Tel.: (88) 3412-1256

TRABALHO DE PROGRAMAÇÃO FUNCIONAL

Avaliação de uma Linguagem Puramente ou Parcialmente Funcional

Equipe: Jonas Lopes - 385433, Francisco Samuel - 383860

Disciplina: Programação Funcional

Professor: Ricardo Reis

Linguagem: Miranda

MIRANDA

Introdução:

Linguagem de programação descrita como preguiçosa, puramente funcional, não estrita (não precisa ou não rigorosa), polimórfica e de alta ordem.

Operacionalmente, uma função estrita é uma função que sempre avalia seus argumentos, logo uma função não-estrita talvez não avalie alguns de seus argumentos.

Projetada por David Turner, da Universidade de Kent, como sucessora de duas outras linguagens:

- SALS (St Andrews Static Language)
- KRC (Kent Recursive Calculator)

Produzida pelo Research Software Ltd. of England e foi a primeira linguagem puramente funcional a ser suportada comercialmente.

“Ainda há muitas questões importantes sem resposta, parece claro que chegamos a um ponto em que é apropriado disponibilizar o que descobrimos até agora para uma comunidade maior, na forma de implementações estáveis de linguagens de programação completas e bem documentadas.” - Esta foi a principal motivação para a criação da Linguagem Miranda.

Histórico:

Lançada primeiramente em 1985, como um interpretador rápido em C para sistemas operacionais Unix, e então, com novos releases em 1987 e 1989.

Miranda teve forte influência posteriormente na linguagem Haskell. Mais tarde a linguagem de programação Haskell surge com muitas semelhanças a Miranda.

Nomenclatura:

- Por que do nome Miranda?

Miranda que no Latim significa “para ser admirado em”. Este nome tem sido utilizado na Inglaterra e na América como nome próprio feminino desde o início do século XVII. O seu primeiro uso como nome próprio a uma menina foi devido a Shakespeare, em sua peça teatral *A Tempestade* (1611).

No ato 5, cena 1 da peça, a personagem Miranda faz um discurso com as palavras: “*O Brave New World!*” (“*Ó Bravo Novo Mundo!*”). A ideia é de que a linguagem Miranda é uma introdução ao Bravo Novo Mundo da programação funcional. E por se tratar de um nome próprio, e não um acrônimo, somente a primeira letra de Miranda está em maiúsculo.

Então Miranda:

- É uma linguagem de programação funcional
- Foi projetada por David Turner, da Universidade de Kent
- De avaliação preguiçosa, tipagem forte, polimórfica e com um poderoso sistema de módulos

Estrutura:

Um programa em Miranda é normalmente 5 a 15 vezes mais curto do que seu correspondente em C ou Java.

O compilador de miranda é incorporado como um sistema interativo implementado sob UNIX fornecendo acesso a um editor de tela.

E as principais características (features) da linguagem Miranda são:

- **Puramente funcional (sem efeitos colaterais).**

(em Ciência da Computação, se diz que uma operação, função ou expressão tem efeitos colaterais se esta, além de retornar um valor, modifica o estado de alguma variável fora do seu ambiente local (seu escopo))

- **Alta ordem (suporte a functional data)**

(Conceito bem simples, em que possui dois tipos de função de alta ordem, o tipo 1 são aquelas funções que recebem uma ou mais funções como argumentos, e o tipo 2 são aquelas funções que devolvem outra função como valor de retorno)

- Avaliação preguiçosa (suporta funções não estritas e objetos de dados infinitos)

(Avaliação preguiçosa (ou call-by-need) é uma estratégia de avaliação que retarda a avaliação de uma expressão até que esta seja necessária, e que também evita avaliações repetidas)

- **Compreensão de listas**

(A compreensão de lista nos provê uma forma simples de criar uma lista baseada em algo iterável. Durante a criação da lista, os elementos desse “algo iterável” podem ser condicionalmente incluídos em uma nova lista e transformados conforme necessário)

- **Tipagem forte polimórfica**

(Usando o polimorfismo, a função ou tipo de dado pode ser escrita genericamente para que possa suportar valores idênticos sem depender de seu tipo. Essas funções e tipos de dados são chamados funções genéricas e tipos de dados genéricos respectivamente e formam a base da programação genérica.)

- **Tipos e módulos abstratos de dados**

(Geralmente, um módulo agrupa vários tipos e funções com funcionalidades relacionadas, caracterizando assim uma finalidade bem definida. Nos casos em que um módulo define um novo tipo de dado e o conjunto de operações para manipular dados desse tipo, falamos que o módulo representa um tipo abstrato de dados. Nesse contexto, abstrato significa “esquecida a forma de implementação”, ou seja, um Tipo Abstrato de dados é descrito pela finalidade do tipo e de suas operações, e não pela forma como está implementado)

O compilador de Miranda é incorporado como um sistema interativo implementado sob UNIX fornecendo acesso a um editor de tela.

Os arquivos Miranda tem a extensão “.m ” – Exemplo: “*questao1.m*” .

Operadores Lógicos / Relacional

```
Miranda (y = 42) & (x = 0) & (y = x)
False
```

```
Miranda (y = 42) & ((x = 0) & (y = x))
False
```

```
Miranda (x = 0) \\/ ((y div x) > 23)
True
```

Table 1.3 Logical operations on Booleans.

~	logical negation
&	logical conjunction
\/	logical disjunction

Tipos de dados:

Os tipos básicos embutidos em Miranda são:

- Números (inteiros de tamanho ilimitado e float de precisão dupla)
- Caracteres
- Listas
- Tuplas
- Funções

Aqui o tipo string é simplesmente uma lista de caracteres, tuplas são sequências de elementos que podem ser de tipos diferentes, assim como em Haskell, e são escritas delimitadas por parênteses.

Tipos definidos pelo usuário com subestrutura arbitrária podem ser introduzidos por tipos algébricos de dados, escrevendo equações de tipo.

Exemplo:

Um exemplo breve do estilo Miranda de programar:

```
primos = sieve [2..]
```

```
where
sieve (p:x) = p : sieve [ n | n <- x ; n mod p > 0 ]
```

a função acima é a definição de uma lista de todos os números primos (perceba que é uma lista infinita) utilizando o Crivo de Eratosthenes

Observe que a expressão entre colchetes é uma compreensão de lista, ou seja *uma lista com n elementos, tal que cada n-ésimo elemento é extraído de x e não é divisível por p*. Aqui o teste de divisibilidade usa o resultado da função *mod*.

Torre de Hanoi:

hanoi.m

```
soln = title++hanoi 8 "A" "B" "C"
title = "SOLUTION TO TOWERS OF HANOI WITH 8 DISCS\n\n"
hanoi 0 a b c = []
hanoi (n+1) a b c = hanoi n a c b
                    ++ move a b ++
                    hanoi n c b a
move a b = "move the top disc from "++a++" to "++b++"\n"
```

Fibonacci:

fibs.m

```
fibs = map fib [0..]
fib 0 = 0
fib 1 = 1
fib (n+2) = fibs!(n+1) + fibs!n

test = layn (map shownum fibs)
```

Fibonacci com lista de compreensão:

```
fibs = [a | (a,b) <- (0,1), (b,a+b) ..]
```

QuickSort:

quicksort.m

```
qsort [] = []  
qsort (a:x) = qsort [b|b<-x;b<=a]  
              ++ [a] ++  
              qsort[b|b<-x;b>a]
```

Referências:

<http://www0.cs.ucl.ac.uk/teaching/3C11/book/Ch1-2.pdf> - Programming w/ Miranda book

[https://en.wikipedia.org/wiki/Miranda_\(programming_language\)](https://en.wikipedia.org/wiki/Miranda_(programming_language))

[https://es.wikipedia.org/wiki/Miranda_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Miranda_(lenguaje_de_programaci%C3%B3n))

<http://miranda.org.uk/>

<http://miranda.org.uk/short.html>

<http://miranda.org.uk/examples>

http://sci-hub.tw/https://link.springer.com/chapter/10.1007%2F3-540-15975-4_26?fbclid=IwAR38M2inLMjfx-aokx1jhQep4YqozBuDfo2qghAKtvUXmOAXN5QRGK_ruLc

Coisas interessantes:

<https://www.cs.kent.ac.uk/people/staff/dat/tfp12/tfp12.pdf>

<https://www.cs.kent.ac.uk/people/staff/dat/miranda/whyfp90.pdf>