CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO
ENGENHARIA DE COMPUTAÇÃO

ANDRÉ MARCELINO DE SOUZA NEVES

Dissertação sobre linguagem de programação Haskell

Professor: Leonardo Lacerda Alves

TIMÓTEO
2018

1 Paradigma de programação

Haskell é uma linguagem de programação que foi nomeada por Haskell Brooks Curry em 1987 (WIKIPéDIA, 2018). É caracterizada por ser puramente funcional, não estrita e baseada no cálculo lambda, além de outras características que a diferencia de outras linguagens consideradas convencionais e de maior popularidade no senso comum (HASKELLWIKI, 2018b), como *Java*, *C++* e *Python*.

O cálculo lambda é um sistema matemático formal criado com o intuito de estudar os aspectos mais básicos dos quais operadores ou funções podem ser combinados para formar outros. "Trata-se de uma linguagem expressiva, a qual é suficientemente poderosa para expressar todos os programas funcionais, e por conseguinte, todas as funções computáveis."(BARRETO, 2018).

Nas linguagens funcionais, existe grande capacidade de abstração, que é basicamente o uso de recursos os quais o funcionamento interno é oculto. Por exemplo: O uso de memória é gerenciado pela linguagem, que aloca armazenamento necessário e desaloca quando for possível. Como a programação é feita em alto nível, existe grande facilidade de escrita e manutenção, em detrimento de o programador ter menos controle da máquina. Nesse sentido, um programa é uma expressão que é executada pela avaliação de sua estrutura, e sua descrição é baseada em *o que* fazer, ao contrário do paradigma imperativo, que exige uma colocação precisa a respeito de *como* uma operação deve ser feita (HASKELLWIKI, 2018b).

2 Conceitos fundamentais

Por ser uma linguagem essencialmente diferente das demais conhecidas comumente, haskell possui diversos conceitos sem os quais não existiria todo o poder oferecido. Alguns estão citados a seguir:

Compreensão de lista *(List Comprehension)*: Baseado nas ideias matemáticas de manipulação de conjuntos, a compreensão de listas é a definição de obter listas de conteúdos específicos a partir de outras. Um exemplo é a definição *[x*2 | x <- [1..10]]*, a qual refere-se aos 10 primeiros pares (LIPOVAčA, 2011a).

Avaliação preguiçosa *(Lazy evaluation)* e linguagem não estrita: Uma linguagem não estrita não exige que os parâmetros reais de uma função sejam completamente avaliadas em um primeiro instante. Isso caracteriza o conceito de avaliação preguiçosa, a qual significa que cada parâmetro de uma função será avaliado (calculado) apenas quando for necessário. Esse recurso permite a definição de estruturas de dados infinitas, como a seguinte, que contém todos os pares: *evens = [2, 4, ...]*. Nenhum computador real seria capaz de processar por completo essa lista, mas com o uso da avaliação preguiçosa, é possível definir uma função que utiliza partes específicas dessa estrutura de dados, e devido à avaliação preguiçosa a linguagem de programação irá processar cada valor apenas no momento que for necessário (SEBESTA, 2011).

Casamento de padrões *(Pattern Matching)*: Casamento de padrões é um recurso utilizado

para definir funções que possuem definições distintas para cada padrão de entrada (LIPOVAčA, 2011b). Um exemplo simples é sua utilização para a definição da função fatorial:

```
factorial :: (Integral a) => a -> a
factorial 0 = 1
factorial n = n * factorial (n - 1)
```

A chamada de *factorial 0* corresponde ao primeiro padrão, enquanto *factorial 6* corresponde ao segundo. Esse recurso poupa trabalho do programador em definições complexas, já que não é necessário preocupar-se com a verificação de padrões, mas a própria linguagem o faz.

3 Identificadores e tipos de dados

Um identificador em Haskell começa por uma letra maiúscula ou minúscula seguidas por outras letras, dígitos ou *underlines*, e não pode ser equivalente a algumas das palavras reservadas da linguagem como *case, class, data, do, else, if, import, entre outras.* É importante considerar que os nomes de identificadores são *case-sensitive* (OLIVEIRA; MORESCHI, 2015).

Haskell é uma linguagem fortemente e estaticamente tipada. Apesar dessa última característica, é possível trabalhar com inferência de tipos, o que significa que caso não seja definido o tipo de uma expressão, o compilador é capaz de reconhecer os padrões e encontrar a opção adequada para a mesma (HASKELLWIKI, 2018a).

Alguns dos principais tipos incluídos na linguagens Haskell são: *Bool, Char, Double, Float, String, Tuple* e *List* (HASKELL, 1998).

4 Adoção comercial, industrial e científica

Haskell é uma linguagem que ainda encontra-se no processo de desenvolvimento, e com isso muitas pesquisas ainda são realizadas sobre a mesma (WIKIPéDIA, 2018). Apesar de ser puramente funcional, um paradigma que de certa forma difere da maneira como a maioria dos programadores estão acostumados a trabalhar, Haskell está presente em alguns importante projetos comerciais e industriais, que podem ser encontrados na página oficial da linguagem: <https://github.com/erkmos/haskell-companies>.

Sigma é um sistema utilizado pelo *Facebook* para combater ações maliciosas como *spams*, ataque de *phishing* e links para *malware's*. O sistema utiliza a linguagem Haskell e é capaz de responder a mais de um milhão de requisições por segundo. (FACEBOOK, 2015)

O serviço online <https://chordify.net/> utiliza a linguagem Haskell para o módulo que transforma um arquivo de música na sequência de acordes correspondentes (HASKELLWIKI, 2014). Detalhes do projeto estão no artigo de Magalhães e Haas (2011).

REFERÊNCIAS

BARRETO, Jorge Muniz. Technical Report, *Cálculo Lambda*. Departamento de Informática e de Estatística - Universidade Federal de Santa Catarina, 2018. Disponível em: <http://www.inf.ufsc.br/~j.barreto/PF/CalLambda.htm>. Acesso em: 15 nov. 2018.

FACEBOOK, Code. *Fighting spam with Haskell*. 2015. Disponível em: <https://code.fb.com/security/fighting-spam-with-haskell/>. Acesso em: 19 nov. 2018.

HASKELL. *The Haskell 98 Report: Predefined Types and Classes*. 1998. Disponível em: <https://www.haskell.org/onlinereport/basic.html>. Acesso em: 16 nov. 2018.

HASKELLWIKI. *Haskell in industry*. 2014. Disponível em: <https://wiki.haskell.org/Haskell_in_industry>. Acesso em: 19 nov. 2018.

____. *Haskell em 10 minutos*. 2018. Disponível em: <https://wiki.haskell.org/Haskell_em_10_minutos>. Acesso em: 17 nov. 2018.

____. *Introduction*. 2018. Disponível em: <https://wiki.haskell.org/Introduction>. Acesso em: 15 nov. 2018.

LIPOVAčA, Miran. Learn you a haskell for great good! - list comprehension. p. 15, 2011. Disponível em: <http://learnyouahaskell.com/starting-out#im-a-list-comprehension>. Acesso em: 15 nov. 2018.

____. Learn you a haskell for great good! - pattern maching. p. 35–36, 2011. Disponível em: <http://learnyouahaskell.com/syntax-in-functions#pattern-matching>. Acesso em: 15 nov. 2018.

MAGALHãES, José Pedro; HAAS, W. Bas de. *Functional Modelling of Musical Harmony - An experience report*. 2011. Disponível em: <http://dreixel.net/research/pdf/fmmh.pdf>. Acesso em: 19 nov. 2018.

OLIVEIRA, Pâmella de; MORESCHI, Vitor de Nardi. *Seminário Linguagens de Programação*. Departamento de Informática - Centro Tecnológico - Universidade Federal do Espírito Santo, 2015. Disponível em: <https://inf.ufes.br/~vitorsouza/wp-content/uploads/teaching-lp-20151-seminario-haskell.pdf>. Acesso em: 17 nov. 2018.

SEBESTA, Robert W. *Conceitos de Linguagens de Programação*. 9. ed. [S.l.]: Bookman Companhia Ed, 2011. 715–716 p. ISBN 9780136073475.

WIKIPéDIA. *Haskell (linguagem de programação)*. 2018. Disponível em: <https://pt.wikipedia.org/wiki/Haskell_(linguagem_de_programa\%C3\%A\%C3\%A3o)>. Acesso em: 16 nov. 2018.