

Gustavo Lopes Rodrigues, Lucas Santiago, Pedro Souza, Thiago Henriques

Uma Introdução a Haskell

Belo Horizonte

2020

Sumário

1	Introdução	1
2	Histórico sobre a linguagem, com sua cronologia	2
3	Paradigma a que pertence	2
4	Características mais marcantes da linguagem	3
5	Linguagem similares ou confrontantes	3
6	Exemplo(s) de programa(s)	4
7	Estudo de Caso, sobre o(s) tema(s)	4
8	Considerações finais	4
Apêndices		5
APÊNDICE A Teste		6
1 Introdução		

Em 1930, Alonzo Church , matemático estadunidense apresentou o Cálculo Lambda, como parte da investigação dos fundamentos da matemática. O Cálculo Lambda é um sistema que estuda funções recursivas computáveis, e foi utilizada como base para as teorias e fundamentos matemáticos por trás do paradigma da Programação Funcional. Ele também pode ser considerado a primeira linguagem programação funcional, todavia, não foi projetada para ser executada em computadores, sendo apenas um modelo que descreve relações entre funções simples, permitindo criar funções mais complexas.

Com o passar dos anos, varias linguagens funcionais foram criadas, sendo alguns exemplos a linguagem LISP em 1955 e a ML no final da década de 70. Porém, não havia um padrão para as linguagens desse paradigma, e quando chegou a segunda metade da década de 80, havia uma necessidade para criar uma única linguagem, que englobasse as melhores práticas de projeto, além de implementar as técnicas funcionais que estavam em alta na época.

Em Setembro de 1987, aconteceu a conferência *Functional Programming Language and Computer Architecture* em Portland - Oregon, para que fosse discutido justamente tais questões.

Dessa conferência, surgiu as seguintes diretrizes:

- Ser viável para o ensino, pesquisa e aplicações, incluindo sistema de larga escala;
- Ser completamente descritiva via publicação no tocante à sua sintaxe e sua semântica;
- Não ser proprietária, tal que qualquer um pudesse implementá-la e distribuí-la;
- Basear-se em ideias que envolvessem o senso comum;
- Reduzir a diversidade desnecessária de outras linguagens funcionais.

Além disso, eles decidiram não começar do zero, e elegeram a linguagem Miranda, como base do projeto, por ser a mais amadurecida e bem projetada entre as demais. E foi a partir dessa linguagem, que nasceu Haskell.

2 Histórico sobre a linguagem, com sua cronologia

A criação do Haskell se deu em 1987, quando Peyton Jones e Paul Hudak fizeram uma reunião no meio da FPCA (Functional Programming and Computer Architecture Conference) para criar uma linguagem de programação funcional e comum.

Como ponto de partida a equipe decidiu por escolher uma linguagem já existia como ponto de partida para seu projeto. A linguagem Miranda foi a escolhida, por ter um design robusto e intuitivo, ela atendia a maioria das metas estabelecidas e além de estar sendo utilizada em sites.

Porém após algumas conversas, David Turner (criador da Miranda) não autorizou o uso da Miranda. Isso se deve pelo fato dos objetivos da equipe serem diferentes dos dele, deixando assim um pedido que fizessem uma linguagem diferente.

Por mais que a Miranda não pudesse ser utilizada, a equipe se inspirou na linguagem e adotou elementos específicos. Sendo assim, a implementação do Haskell começou do zero, permitindo assim o desenvolvimento mais fluido da linguagem e a criação de funções únicas.

The Yale Meeting foi a primeira reunião presencial que ocorreu logo após a reunião FPCA, no qual foi decidido os principais objetivos que a linguagem proporcionaria, como também a escolha do nome para a nova linguagem. Segue as metas estabelecidas na reunião:

- Ser viável para o ensino, pesquisa e aplicações, incluindo sistema de larga escala;
- Ser completamente descritiva via publicação no tocante à sua sintaxe e sua semântica;
- Não ser proprietária, tal que qualquer um pudesse implementá-la e distribuí-la;
- Basear-se em ideias que envolvessem o senso comum;
- Reduzir a diversidade desnecessária de outras linguagens funcionais.

3 Paradigma a que pertence

O paradigma oferece e determina a visão que o programador possui sobre a estruturação e a execução do programa. Um exemplo bem famoso de paradigma é o POO (Programação orientada a objetos). Já a Programação funcional é um paradigma que descreve uma expressão matemática a ser avaliada, mapeando dos valores de entradas nos valores de retorno, por meio de funções. Em outras palavras: a programação funcional só funciona em cima de funções.

Eis mais algumas características do paradigma

- Dados imutáveis e são evitados estados.
- Não existe efeitos colaterais
- "Funções puras"(sem efeitos colaterais)
- Cálculo Lambda

4 Características mais marcantes da linguagem

- Como dito anteriormente, a linguagem só faz utilização de funções e funções dentro de funções. Por isso Haskell é descrito como puramente funcional.
- Haskell possui uma sintaxe simples, elegante e concisa. Como resultado, programas em Haskell possuem poucas linhas.
- Além disso, a linguagem usa avaliação preguiçosa(Lazy evaluation), que é uma técnica para atrasar a computação até um ponto em que o resultado da computação é considerado necessário.
- Tipagem estática: Verificação dos tipos usados em dados e variáveis para garantir que sempre está sendo usado um tipo que é esperado em todas as situações.
- função de ordem superior: Função que tem como argumento uma outra função, ou que produz uma função como resultado.

5 Linguagem similares ou confrontantes

- Prolog
- LISP
- Scheme
- ML
- Miranda
- Elixir

6 Exemplo(s) de programa(s)

7 Estudo de Caso, sobre o(s) tema(s)

8 Considerações finais

Apêndices

APÊNDICE A – Teste