


Prof. Valdigleis S. Costa

Professor Adjunto, Colegiado de Ciência da Computação
Universidade Federal do Vale do São Francisco

valdigleis@gmail.com

MATEMÁTICA DISCRETA

Um curso moderno

The bottom half of the page features two large, overlapping geometric shapes. On the left is a large red triangle pointing towards the bottom right. On the right is a large orange triangle pointing towards the bottom left. These two triangles meet at a diagonal line that runs from the middle of the left edge to the bottom right corner, creating a dynamic, abstract background.

Copyright © 2019-2024 Valdigleis S. Costa

Este texto NÃO possui qualquer tipo de vínculo editorial, e não possui fins lucrativos.

Página pessoal do autor <https://valdigleis.site>



Este material é licenciado sob a Licença Atribuição-NãoComercial-CompartilhaIgual 3.0 Não Adaptada (CC BY-NC-SA 4.0). Você pode obter uma cópia da licença acessando a página:

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.pt>

ou enviando uma carta para Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Este tomo foi escrito com base em uma coleção de notas de aulas do autor, o mesmo foi redigido usando um *template* desenvolvido pelo próprio autor. Este texto foi escrito com o conjunto de macros L^AT_EX (em sua versão 2) e compilado usando as ferramentas LuaL^AT_EX e BibT_EX, tais ferramentas fornecidas pelas distribuições T_EXLive e MacT_EX, respectivamente nos sistemas operacionais *Unix-like*: Gnu/Linux e no Mac OS X, para edição foram usados os *softwares* livres de edição textual NeoVim (versão 0.10.1), além disso, o sistema de controle de versão adotado é o Git (versão 2.34.1).

Release compilado em 28 de outubro de 2024 (880 minutos após a meia-noite).



Sobre este documento

Há muito tempo, em uma galáxia muito, muito distante, quando eu era aluno de graduação, as disciplinas ligadas à grande área chamada de matemática discreta eram, em geral, lecionadas com um enfoque totalmente formal e analógico¹. Para mim, isso era maravilhoso, pois amo matemática, entretanto, o tempo passou, o mundo mudou e hoje sou o professor e não o aluno. Hoje, a mentalidade e os interesses dos alunos são diferentes dos que eu tinha, há uma necessidade de ver as coisas visualmente, na prática, o mundo abstrato da matemática não tem mais tanto charme quando tinha na minha época, por esse motivo, decide escrever esse material.

Meu objetivo para este material, é que ele seja, uma fonte de consulta para aprender os tópicos da matemática discreta de forma interativa, usando a linguagem Haskell como ferramenta para implementações práticas dos tópicos. Entre esses tópicos estarão:

- (i) Conjuntos, relações e funções;
- (ii) Introdução à Lógica;
- (iii) Teoria dos códigos;
- (iv) Linguagem formais e autômatos;
- (v) Computabilidade e decidibilidade;
- (vi) Teoria de Grafos;
- (vii) Tipos abstrados de dados;
- (viii) Fundamentos de Categorias;

No que diz respeito a organização deste documento, ele irá iniciar apresentado conceito básicos e fundamentais da linguagem Haskell, depois serão apresentados os tópicos listados anteriormente, intercalados (quando necessário) com apresentações adicionais de outros conceitos de nível intermediário e avançado da linguagens Haskell, nessa primeira parte do documento será assumido que o leitor não tem qualquer tipo de conhecimento prévio sobre a linguagem Haskell, os leitores que por ventura já sejam maduros na linguagens, podem simplesmente pular essa primeira parte do documento.

Espero que esse material seja útil para você caro leitor, e se possível, gostaria de solicitar sua ajuda para melhora cada vez mais este material! Pois, uma vez que, este material ainda é um projeto em andamento e possivelmente sua escrita nunca será realmente concluída com total aprovação minha, tenho certeza que você poderá encontrar diversos erros, que gentilmente gostaria de solicitar que você leitor me envie por e-mails² ou *issues*³ reports de tais erros, no caso de ser meu aluno também pode fazer apontamentos através da comunidade **extra-classe**⁴. Também, agradeço por qualquer sugestão para melhorar o texto.

¹Tudo era feito com lápis, papel e borracha.

²E-mail do autor: valdigleis@gmail.com

³Páginas de *issues*: <https://github.com/valdigleis/mdm/issues>

⁴Acessível através do link <https://valdigleis.site/extraclasse>

Sumário

I Uma introdução sucinta para Haskell

1	Conhecendo Haskell	3
1.1	O que é Haskell?	3
	Referências Bibliográficas	5

Parte I

Uma introdução sucinta para Haskell

Conhecendo Haskell

*“-Comece pelo começo”, disse o
Rei de maneira severa,
“-E continue até chegar ao fim,
então pare!”*

*Lewis Carroll, Alice no País das
Maravilhas.*

1.1 O que é Haskell?

Haskell como dito em [1], é uma linguagem de programação funcional, pura, de alto nível e para propósito geral¹. O projeto que deu origem a linguagem haskell remonta ao final dos anos 1980 e início dos anos 1990, a linguagem recebeu seu nome para em homenagem o lógico e matemático americano, Haskell Curry (1900–1982), a linguagem foi desenvolvida por um comitê com o objetivo de unificar várias linguagens funcionais da época, como Miranda e ML [3]. O objetivo grandioso era criar uma linguagem funcional que fosse ser o padrão mais adequado possível, tanto para o meio de pesquisa (acadêmico) quanto para o ambiente de desenvolvimento prático (coorporativo e de mercado).

A linguagem Haskell combina diversas características presentes em outras linguagens com características próprias. Uma característica famosa usando em Haskell (mas que não foi criado em Haskell), é por exemplo, o uso de “*lazy evaluation*” (avaliação preguiçosa em português), tal característica faz com que as expressões sejam avaliadas (ou computadas) apenas quando realmente forem necessárias, o que possibilita entre outras coisas, ser possível criar estruturas de dados capazes de representar quantidades infinitas de dados, essa característica também possibilita que seja possível otimiza o uso da memória. Já do ponto de visto particular de Haskell, ela foi a linguagem responsável por introduz conceitos como “monads” para

¹Famosa inclusive por ser a base do sistema de *tiling windows manager* chamado Xmonad, detalhes em <https://xmonad.org/>.

as linguagens de programação², os monads são estruturas que permitem lidar de forma pura e controlado, com as operações sequenciais e os dispositivos de I/O, mantendo o código limpo e modular.

²O conceito de matemática monads já era conhecido antes de sua implementação na linguagem Haskell, detalhes em [2].

Referências Bibliográficas

- [1] M. Lipovaca. *Learn you a haskell for great good!: a beginner's guide*. no starch press, 2011.
- [2] A. S. Mena. *Beginning Haskell: A Project-Based Approach*. Apress, 2014.
- [3] D. Oleynikov and G. Dreimanis. Haskell. History of a Community-Powered Language. Acessado em 28 de Outubro de 2024 na página <https://serokell.io/blog/haskell-history>, 2024.