



On Understanding Types, Data Abstraction and Polymorphism

Rafael Castro G. Silva

rafaelcgs10@gmail.com

Departamento de Ciência da Computação
Centro de Ciências e Tecnológicas
Universidade do Estado de Santa Catarina

2 de Setembro de 2017

Classificação de Sistemas de Tipos

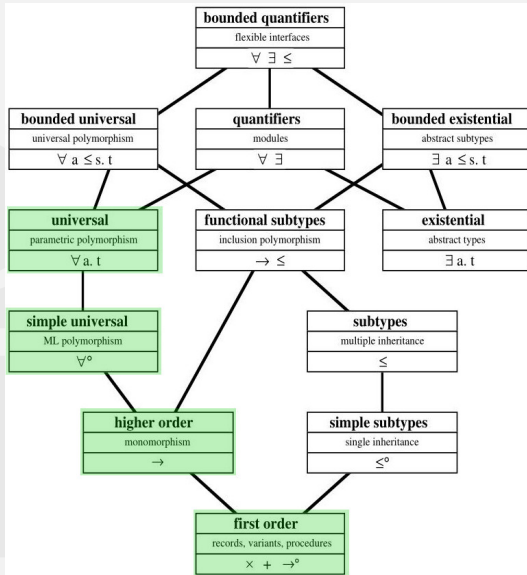


Figura: Classificação Sistemas de Tipos. Retirado de [1].



Primeira Ordem

$FunType := ConstType \rightarrow FunType$
 $ConstType := Int, Float, String...$

- Funções não são dados.
- Linguagens: Fortran, Java < 8, Python < 2, Ruby...



Ordem Superior

$Type := Type \rightarrow Type | ConstType$
 $ConstType := Int, Float, String...$

- Funções são dados.
- Linguagens: Algol 68, C, Pascal, Java ≥ 8 , Python ≥ 2 , Haskell...



Polimorfismo ML (Simple Universal)

$$PolyType := \forall VarType. PolyType \mid Type$$
$$Type := Type \rightarrow Type \mid ConstType \mid VarType$$
$$ConstType := Int, Float, String \dots$$
$$VarType := a, b, c \dots$$

- Funções são polimórficas: uma função assume vários tipos.
- Funções de ordem superior tratam argumentos de maneira monomórfica.
- Linguagens: ML, OCaml, Haskell...



Segunda Ordem (Universal)

$$Poly := \forall VarType. Type | Type \rightarrow Type | ConstType | VarType$$
$$ConstType := Int, Float, String...$$
$$VarType := a, b, c...$$

- Funções são polimórficas: uma função assume vários tipos.
- Funções de ordem superior tratam argumentos de maneira polimórficas.
- Linguagens: Haskell...



Referências

[1] On Understanding Types, Data Abstraction and Polymorphism,
Luca Cardelli and Peter Wegner.