Compilador da Linguagem Funcional Orientada por Objetos *SCRIPT* para *HASKELL*

Iniciação Científica - DCC/UFMG

Wendell Figueiredo Taveira

Orientadora: Mariza Andrade da Silva Bigonha

Setembro de 1999

Problema Resolvido

Compilar programas escritos em *SCRIPT*

SCRIPT é uma linguagem funcional que visa prover uma notação adequada para formular descrições de semântica denotacional.

SCRIPT está inserida em um ambiente cujo objetivo será oferecer ao usuário ferramentas para diminuição do esforço de composição dessas descrições.

... Problema Resolvido

Este ambiente visa o aumento na qualidade das descrições, ou seja, definições mais confiáveis e fáceis de ler.

Útil para estudos, formulações de notações e testes de descrições denotacionais.

Definições de sua gramática e sintaxe já especificadas, sendo necessária a construção do compilador para aplicação prática.

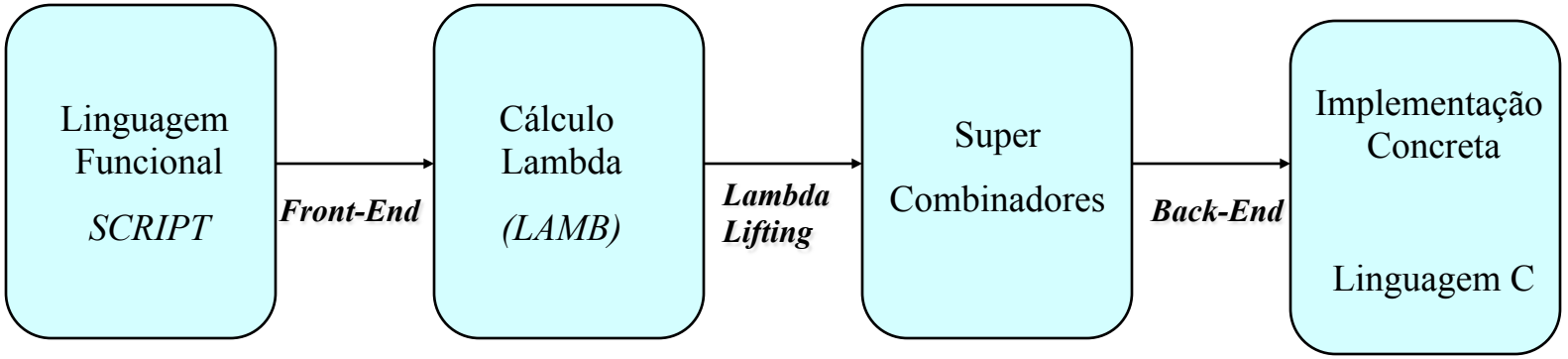
Visão Geral do Compilador *SCRIPT*

- Front-End:
  - análise léxica
  - análise sintática
  - criação e geração da tabela de símbolos
  - verificação de tipos
  - geração do código intermediário *LAMB*

... Visão Geral do Compilador *SCRIPT*

- Fase Intermediária (λ-lifting):
  - tradução de expressões lambda para supercombinadores implementação de otimizações.
- Back-End:
  - compilação de *LAMB* para a linguagem C.

Visão Geral do Compilador *SCRIPT*



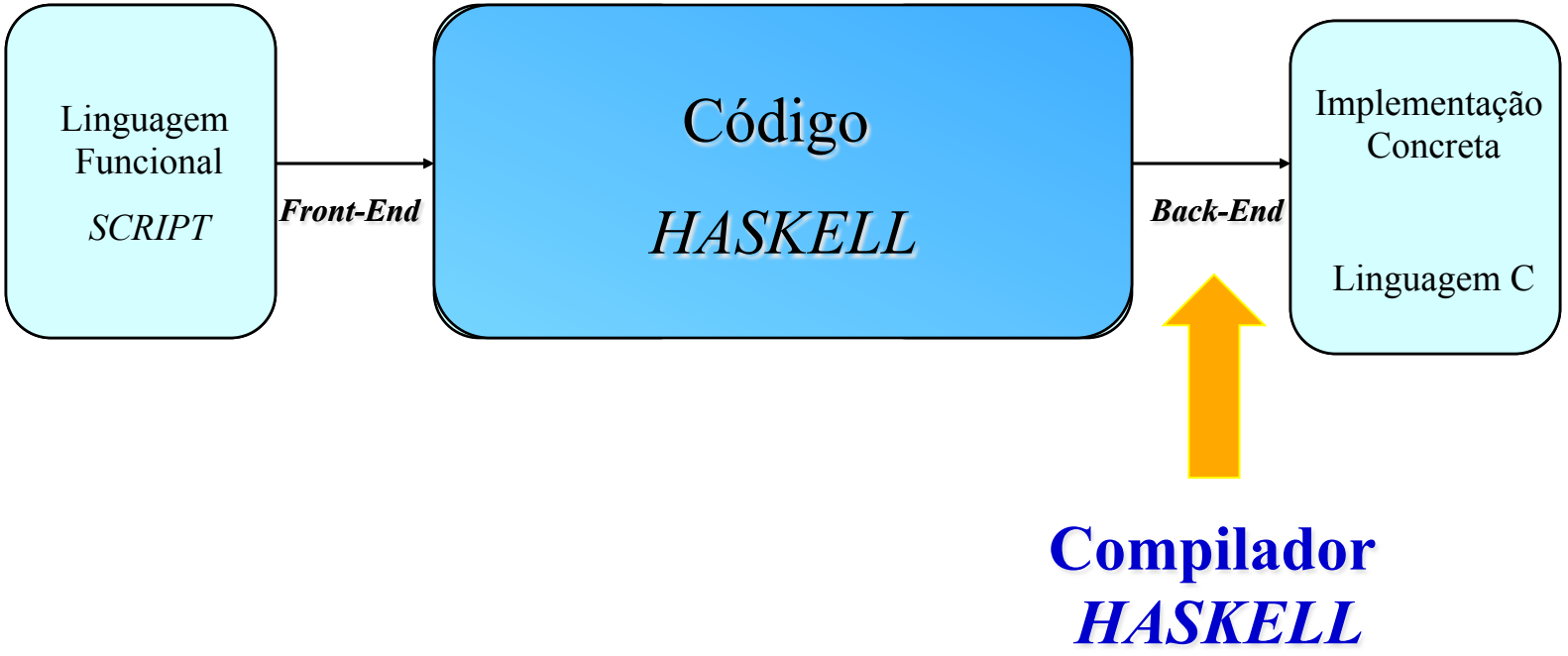
Solução Alternativa Adotada

A abordagem adotada consistiu em:

- Alterar o Front-End do compilador *SCRIPT*:
  - gerar código direto de *SCRIPT* para *HASKELL* ao invés de *LAMB*.
- Utilizar compiladores *HASKELL* como Back-End.

Vantagem: disponibilização, de forma rápida e eficiente, um compilador de *SCRIPT* sem a necessidade de implementar o gerador de código executável.

... Solução Alternativa Adotada



A linguagem Fonte: *SCRIPT*

Características da linguagem *SCRIPT*:

- linguagem funcional orientada por objetos
- usada para definições de semântica denotacional de linguagens de programação
- criada em 1995, pelo professor Roberto da Silva Bigonha

... A linguagem Fonte: *SCRIPT*

Características funcionais da linguagem *SCRIPT*:

- avaliação *lazy*, não estrita
- casamento de padrões
- abstração de funções
- compreensão de listas
- funções de ordem superior

... A linguagem Fonte: *SCRIPT*

Características de OO da linguagem *SCRIPT*:

- classes e objetos
- herança de tipos
- encapsulação de dados
- polimorfismo
- funções virtuais
- ligação dinâmica

A linguagem Objeto: *HASKELL*

Características da linguagem *HASKELL:*

• linguagem de programação avançada e moderna suportando, inclusive, o estilo de OO.

• compiladores e interpretadores eficientes disponíveis.

• um programa *HASKELL* é uma coleção de *scripts* contendo definições de tipos, variáveis, construtores, módulos e classes.

• linguagem fortemente tipada.

Implementação do Compilador *SCRIPT*

- **Esquemas de tradução:**
  - mapeamento das construções de *SCRIPT* para *HASKELL*.
- **Especificação das rotinas semânticas:**
  - manipulação dos atributos associados aos símbolos
- **Implementação do Gerador de Código *HASKELL*:**
  - ***YACC***

... Implementação do Compilador *SCRIPT*

Esquemas de Tradução:

- mapeamentos diretos:

  lista em ***SCRIPT***: <1,2,3>

  lista em ***HASKELL***: [1,2,3]

- mapeamentos não-diretos:

  SCRIPT: (“casa”, 1, TT) EL 2 = 1

  HASKELL: LET (a,b,c) IN (“casa”, 1, True) = b

... Implementação do Compilador *SCRIPT*

Esquemas de Tradução:

- mapeamentos não possíveis de serem construídos:
  - funções virtuais: a verificação de tipos da linguagem objeto *HASKELL* inviabilizou a elaboração do esquema

... Implementação do Compilador *SCRIPT*

Rotinas Semânticas e Gerador de Código

Diferenças entre *LAMB* e *HASKELL*:

- informações ausentes: tabela de símbolos
- ordem do código: adequar a distribuição das produções de modo a gerar código *HASKELL* correto.

Conclusão

- implementado na linguagem C para Unix
- ferramentas: Lex e Yacc
- a utilização de compilador *HASKELL* disponível diminuiu o esforço para obtenção de código executável
- compilador limitado pelas características da linguagem *HASKELL*