



Programming Language Laboratory

Pesquisa em Linguagens de Programação

Fernando Magno Quintão Pereira

- Laboratório de Linguagens de Programação -



`fpereira@dcc.ufmg.br`

Nosso objetivo é tornar os programadores mais produtivos

- Paralelização automática de programas.
- Geração de programas corretos e seguros.



Hall, Padua e Pingali

“Compiler Research: The Next Fifty Years”

Communications of the ACM 2009

fpereira@dcc.ufmg.br

Quem Somos



Roberto Bigonha

Semântica formal

`bigonha@dcc.ufmg.br`

Mariza Bigonha

Compiladores

`mariza@dcc.ufmg.br`



Carlos Camarão

Sistemas de tipos

`camarao@dcc.ufmg.br`

Fernando Pereira

Compiladores

`fpereira@dcc.ufmg.br`



O Laboratório de Linguagens de Programação

- ICEx, sala 3054
- Seminários toda sexta
 - Próxima sexta: “*fun with visitors in Java and Python*”, às 14h00, na 3054
- Seis professores (ES + LP)
- Uma pos-doc
- Quatro alunos de doutorado
- Sete alunos de mestrado
- Quatro alunos de IC



Prêmios ganhos em 2010

- Melhor artigo **SBLP 2010** “*Removing Overflow Tests via Run-Time Partial Evaluation*, Rodrigo Sol, Fernando Pereira e Mariza Bigonha”.

- Segundo melhor artigo

- Terceiro melhor artigo...



- Melhor artigo **SBAC-PAD 2010** “*Performance Debugging of GPGPU Applications with the Divergence Map*, Bruno Coutinho, Diogo Sampaio, Fernando Pereira e Wagner Meira”.



Semântica Formal

- Como descrever o **significado** de um programa de forma **precisa**?

what's the meaning
of this?!



oh, it's a
pronoun



Semântica Formal

- Por que descrever o **significado** de um programa de forma **precisa** é importante?
 - Provar certas propriedades, por exemplo, este programa termina.
 - Gerar compiladores automaticamente.
- Como criar e compor estas descrições de forma **modular**?



Sistemas de Tipos

- Carlos Camarão é membro do comitê de **Haskell** (2011).
 - Este comitê toma as principais decisões sobre o projeto da linguagem.

Haskell λ
A Purely Functional Language





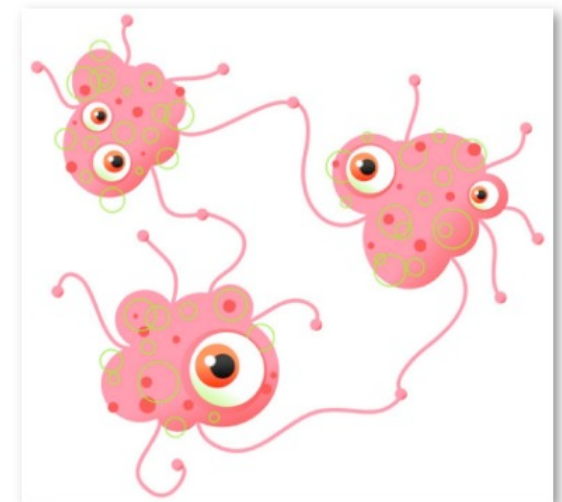
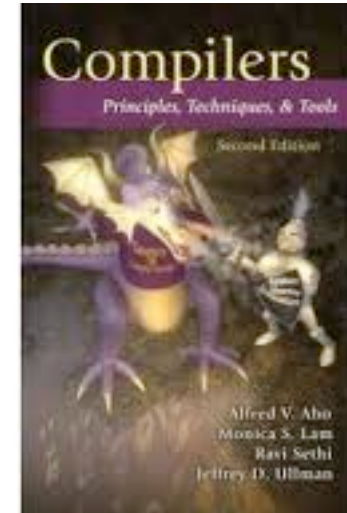
Sistemas de Tipos

- Tipos são uma forma de documentação.
 - Que o compilador consegue verificar!
- Tipos permitem criar programas mais seguros e eficientes.
- Sistemas de tipos nos permitem provar propriedades sobre os programas.
 - Exemplo: este programa nunca lê posições de memória que ele não “possui”.



Conectividade de Software

- Programas crescem muito – de forma quase orgânica
 - Novos módulos são criados
 - Módulos antigos são apagados
- Como entender este crescimento formalmente?
 - Qual o **custo** de modificar uma parte do software?





Geração e Otimização de Código

- O objetivo da pesquisa em compiladores é permitir que programas escritos em linguagens de alto nível usufruam do hardware de forma eficiente.
- Muitos trabalhos com a comunidade de **software livre**.





Geração de Código para CUDA

- Placas de processamento gráfico (GPUs) movimentam uma indústria de milhões de dólares.
- Centenas de threads em um único processador.
- Patches disponíveis no compilador aberto **Ocelot**.



`fpereira@dcc.ufmg.br`



Compilação de Trilhas

- Tornar mais eficiente a execução de programas *JavaScript*.
- Cooperação direta com desenvolvedores do *Mozilla Firefox*.
- Segundo ano do projeto.





Detecção de vulnerabilidades

- Encontrar falhas de segurança via análise estática de programas.
- Primeiro sucesso: 32 bugs em 21 programas PHP de uso industrial.
- Análise disponível em compilador aberto **PHC**.



Identify and Quantify System Vulnerabilities



Software Livre

Google




- 6 Summer of Codes já aceitos.
 - LLVM, Mozilla, GHC
- Patches em cinco compiladores diferentes.
 - TraceMonkey foi baixado 4 milhões de vezes durante a liberação de Firefox 4!





Programming Language Laboratory

- <http://www.dcc.ufmg.br/llp>
- Sala 3054, ICEX
- Seminários bissemanais.
 - Consulte o calendário na página



**Se estiver procurando
um projeto de
pesquisa, vem falar
com a gente!**

fpereira@dcc.ufmg.br