Overview of ICFP 2015

Alejandro Gómez-Londoño

EAFIT University

October 23, 2015
ICFP 2015
Program

Sessions:
- Compilers
- Types
- Foundations
- Cost analysis
- Theorem provers
- Parallelism
- Information flow
- Domain-specific languages
- Data structures
- Contracts
- Type checking
Affiliated events:

- Haskell Implementors Workshop
- Ally Skills Tutorial
- Programming Languages Mentoring Workshop
- Workshop on Functional High-Performance Computing
- Haskell Symposium (2 Days)
- Commercial Users of Functional Programming (3 Days)
- ML Family Workshop
- OCaml Workshop
- Scheme and Functional Programming Workshop
- Functional Art, Music, Modeling and Design
Tracking the flow of ideas through the programming languages literature, Michael Greenberg, Kathleen Fisher, and David Walker
Domain Specific Language "is a computer language specialized to a particular application domain."

---

\(^2\)Wikipedia contributors, Domain-specific language, October 20, 2015.
SQL:

```sql
SELECT id, student
FROM students
WHERE grade > 3.0;
```
Just In Time compiler

Source code -> Code generator -> Native Code

Topics
JIT
“Refinement types allow us to decorate types with logical predicates (think boolean-valued Haskell expressions) which constrain the set of values described by the type.”

```haskell
{-@ divide :: Int -> { v: Int | v != 0 } -> Int @-}
divide :: Int -> Int -> Int
divide n d |
v            = n 'div' d
```

---

3Ranjit Jhala, LiquidHaskell, October 20, 2015.
“The contract system guards one part of a program from another.”

```
#lang racket
(provide (contract-out
    [deposit (-> number? any)]
    [balance (-> number?)])

(define amount 0)
(define (deposit a) (set! amount (+ amount a)))
(define (balance) amount)
```

---

4 racket-lang contributors, Contracts, October 20, 2015.
“Program synthesis is the task of automatically discovering an executable piece of code given user intent expressed using various forms of constraints such as input-output examples, demonstrations, natural language, etc.”

---

Program Synthesis: Opportunities for the next Decade
Ras Bodik, University of Washington

Pycket: A Tracing JIT For a Functional Language
Spenser Bauman, Indiana University

Bounded Refinement Types
Niki Vazou, UC San Diego
Highlights

- **Learning Refinement Types**
  He Zhu, Purdue University

- **An Optimizing Compiler for a Purely Functional Web Application Language**
  Adam Chlipala, MIT CSAIL

- **1ML - Core and modules united (F-ing first-class modules)**
  Andreas Rossberg, Google
Highlights

- Algebras and Coalgebras in the Light Affine Lambda Calculus
  Marco Gaboardi, University of Dundee

- Elaborating Evaluation Order Polymorphism
  Joshua Dunfield, University of British Columbia

- Hygienic Resugaring of Compositional Desugaring
  Justin Pombrio, Brown University