Implementing Functional Dependencies

KATHOLIEKE UNIVERSITEIT
LEUVEN

FACULTEIT
INGENIEURSWETENSCHAPPEN

Master of Engineering: Computer Science

Master's thesis
Michiel Derhaeg

Promotor
Prof. Tom Schrijvers

Academic Year
2017-2018

Main Goal

Implementation of type inference and elaboration into System Fc [3] for Functional Dependencies [1]

Motivation

Example

```
class Coll c e | c → e where
  sing :: e → c
```

Ambiguity

```
sing2 :: (Coll c₁ e, Coll c₂ c₁)
      => e → c₂
sing2 x = sing (sing x)
```

Static Enforcement

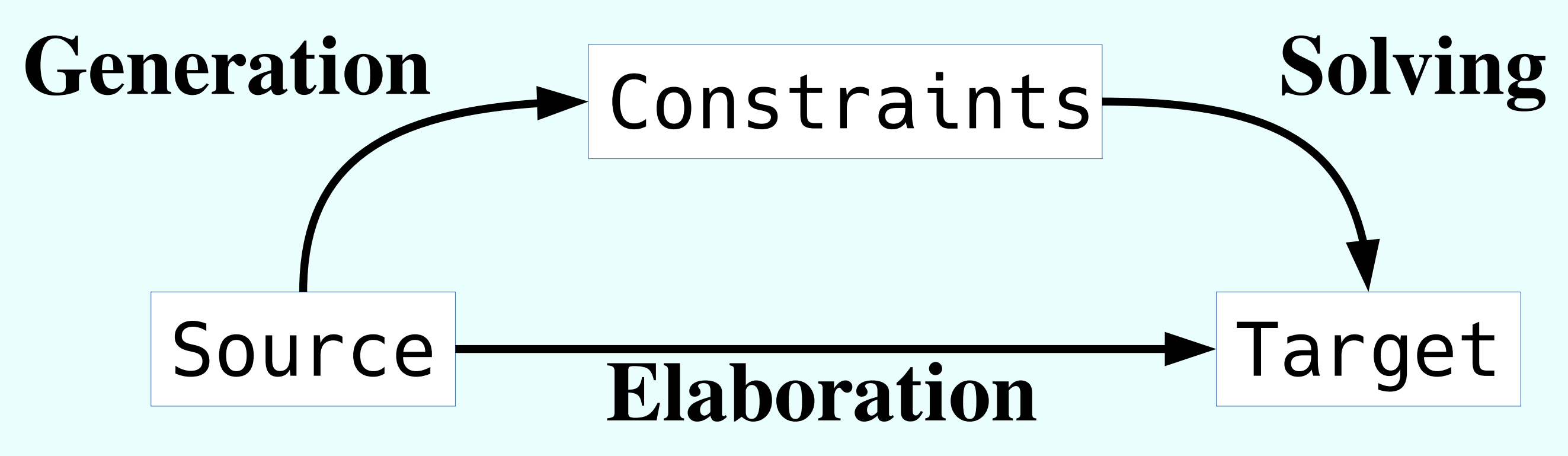
```
instance Coll ByteArray Byte
instance Coll ByteArray Bit
```

Type-level Functions

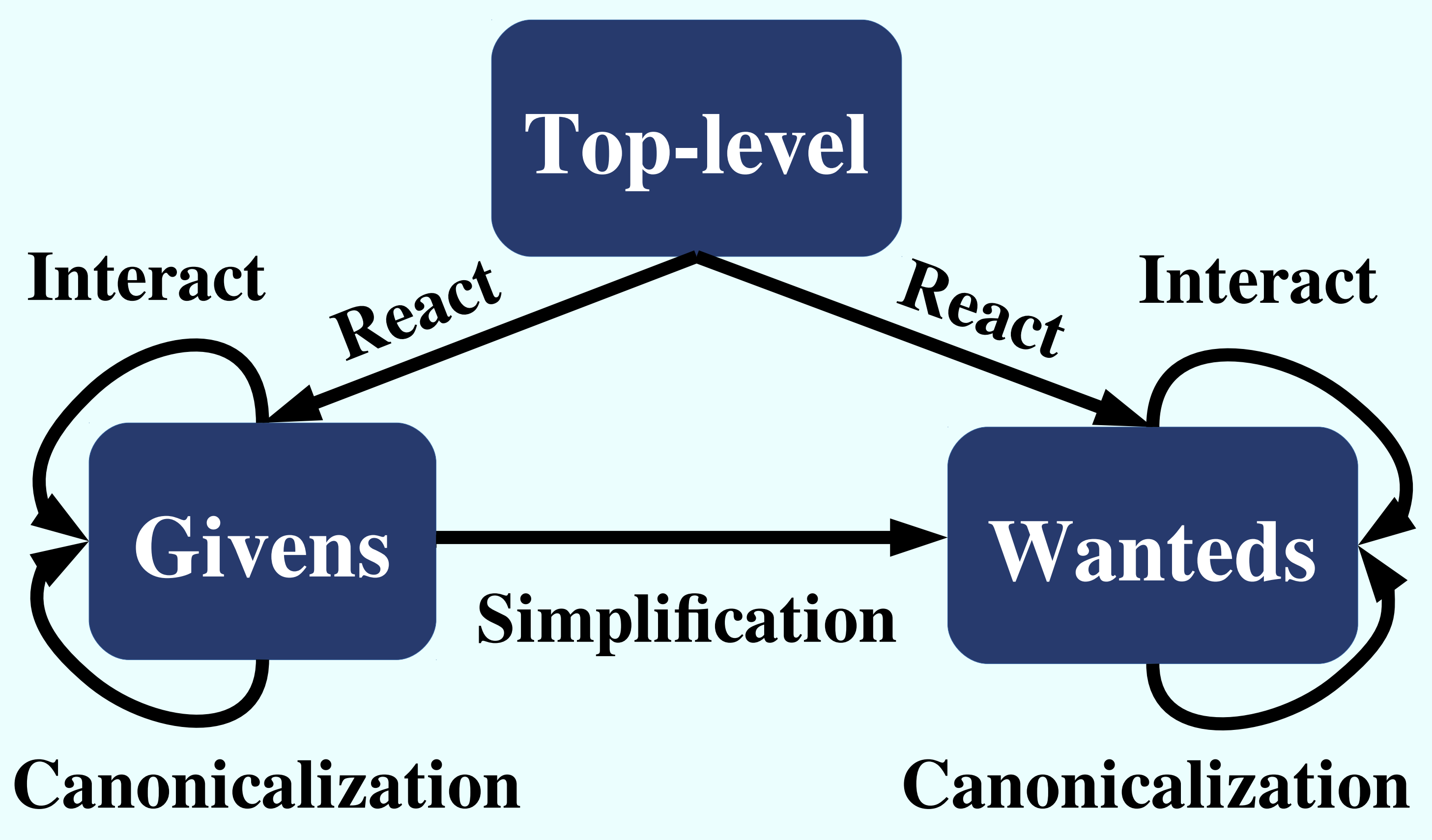
```
class C a b | a → b
instance C Int Bool

f :: C Int b => b → Bool
f x = x
```

General Strategy

- Algorithm by G. Karachalias and T. Schrijvers [3]
- Solving using OutsideIn(X) [4]



OutsideIn(X) Overview



Constraint Solver

Simplification

```
Given:   a = Int
Wanted: Eq a          ──▶  Wanted: Eq Int
```

Interaction

```
Wanteds: Eq Bool
         Eq Bool      ──▶  Wanted: Eq Bool
```

Canonicalization

```
Wanted: [a] = [b]     ──▶  Wanted: a = b
```

Top-level Reaction

```
Top: Eq a => Eq [a]
Wanted: Eq [Int]      ──▶  Wanted: Eq Int
```

Results

- Evaluation of "Elaboration on Functional Dependencies"
- Prototype implementation of Haskell with Functional Dependencies
- Integration of OutsideIn(X) with elaboration into System Fc

[1] Type Classes with Functional Dependencies, 2000, M. P. Jones
[2] System F with Type Equality Coercions, 2011, M. Sulzmann, M. Chakravarty, S. P. Jones, and K. Donnelly
[3] Elaboration on Functional Dependencies, 2017, G. Karachalias and T. Schrijvers
[4] OutsideIn(X), 2011, Dimitrios Vytiniotis, S. P. Jones, T. Schrijvers, and M. Sulzmann