Towards Using Constructive Type Theory for Verifiable Modular Transformations

Steffen Zschaler, Iman Poernomo, Jeffrey Terrell

FREETCO’11
Lancaster, 26 July 2011
Motivation

Object-Relational Mappings

– Classes & Attributes:
  • A Table for every Class
  • A column for every attribute

– Associations
  • Model with foreign keys
  • Many-to-many: Additional table
  • One-to-one: May merge tables

– Inheritance
  • Model with foreign keys \(\rightarrow\) use joins for every query
  • Model with one global table \(\rightarrow\) denormalisation

Would like choice
\(\rightarrow\) Require modular transformation
\(\rightarrow\) Require contracts to make composition safe
Constructive Type Theory 101

\[ a : T \]

- **Example types**
  - Basic types: int, String, ...
  - Function types: \( T_1 \to T_2 \)
  - Dependent types:
    - \( \forall x : T_1. T_2(x) \)
    - \( \exists x : T_1. T_2(x) \)

- **Curry–Howard isomorphism:**
  - Can interpret instances of types as proofs
Model Transformations

• Transform source models into target models
  – Source model instance of MMS
  – Target model instance of MMT

• Can specify as a forall-exists type:
  – $\forall s : \text{MMS}. \text{Pre} s \rightarrow \exists t : \text{MMT}. \text{Post} s t$
Composing Transformations

∀ trans : MMS → MMT → Prop.
∀ proof : (∀ s: MMS. ∃ t: MMT. trans s t ∧ ...).

∀ s: MMS. Pre s → ∃ t : MMT. Post s t ∧ trans s t

Formal description of dependencies between transformations – contract
Structure of Contracts

Three types of statements

1. Requirements on what \( trans \) must ensure
   • Conjoin statements that refer to properties of \( t \) and no further \( \forall \) clauses
   \[ \forall s: \text{MMS.} \exists t: \text{MMT.} \ trans s \ t \land \ SID s = \ TID t \]

2. Requirements on what \( trans \) must not constrain
   • Conjoin statements that equate properties of \( t \) with all-quantified variables
   \[ \forall s: \text{MMS.} \ \forall n: \text{NAT.} \ \exists t: \text{MMT.} \ trans s \ t \land \ TID t = n \]

3. Assumptions \( trans \) may make
   • Precedent of an implication
   \[ \forall s: \text{MMS.} \ \exists t: \text{MMT.} \ assume s \ t \ \rightarrow \ trans s \ t \]
Conclusions & Outlook

• Contracts are important to make composition safe
  – Higher-order type theory can be used to express such contracts for transformations

• Initial work, more research needed
  – Larger, more complex examples
  – Different composition strategies
  – Integration with main-stream transformation languages
THANK YOU FOR YOUR ATTENTION!