Transformation of Models Containing Uncertainty
Michalis Famelis, Rick Salay, Alessio Di Sandro, Marsha Chechik
{famelis, rsalay, adisandro, chechik}@cs.toronto.edu
University of Toronto

Uncertainty in software development
- Many design alternatives
- Incomplete information
- Conflicting stakeholder opinions
- Uncertainty about the content of the model.

Transformations
- Conflicting restrictions to the set of Java (Z3 input strings)
- MDE practitioners use a variety of MTA
- The transformations assume inputs that don't contain uncertainty
- But only too often models contain uncertainty:

Model Transformations With Graph Rewriting
- Encapsulate variable refactoring
- Make fields private and add get/set methods unless they belong to inner class

Intuition (And definition of correctness)

Technique
1. Step 1: Determine applicability
   - (a) Find rule
   - (b) Make sure the rule applies to at least one concretization
   - (c) Copy over unchanged parts
   - (d) Perform additions and deletions

Step 2: Transform graph
- Added and deleted elements become Maybe

Step 3: Transform formula
- Concretizations: ways to resolve uncertainty

Analysis
- Proof of correctness
  - The lifting algorithm implements our intuition
- Proofs of preservation of properties:
  1. Confluence
     - The result of applying a set of rules to a model is in the same regardless of the order of application or the order of matching sites.
  2. Termination
     - Repeated applications will reach a point where the rule will no longer be applicable.

Tool Support
- Reuse partial model implementation in MMTF
- Algorithm implementation
  1. Determine rule applicability
  2. Transform the graph
  3. Transform the formula
     - Java (Z3 input strings)

Case Study
- Object-relational mapping (ORM)
  - “Translate a class diagram to a relational database schema.”
- Triple graph grammar with 5 layered graph rules [Varm06]
- Input model: the Ecore metamodel
- ORM for Ecore is important of CDO and Tenex
- Manually injected points of uncertainty to create partial models with increasing numbers of concretizations
- RO: How does lifting scale with increasing uncertainty?
- Variad: number of concretizations of input
- Measured: time to complete the ORM transformation

References

Next Steps
- Implement lifted semantics as a higher-order transformation (HOT)
  - Given a graph rewrite rule, produce a grammar that implements the lifted semantics
- Expand lifting for other types of model uncertainty, based on the rich MAVO framework [FASE12]