The Semantics of Partial Model Transformations



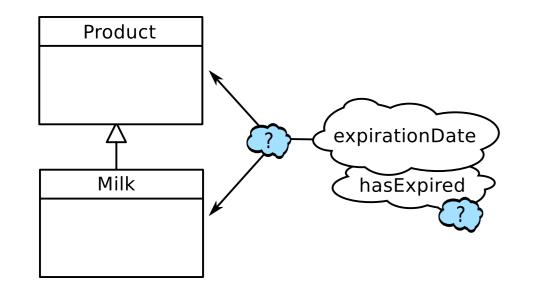
Michalis Famelis, Rick Salay and Marsha Chechik University of Toronto, Canada {famelis, rsalay, chechik}@cs.toronto.edu



Problem Statement

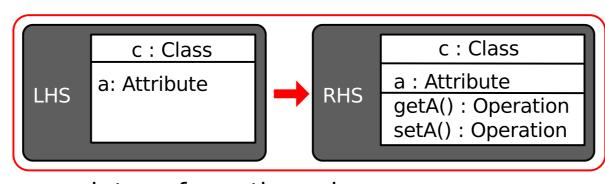
Models with uncertainty

- Represent choice among many possibilities
- Can be refined to many different classical models



Our Goal

Directly transform models containing uncertainty.

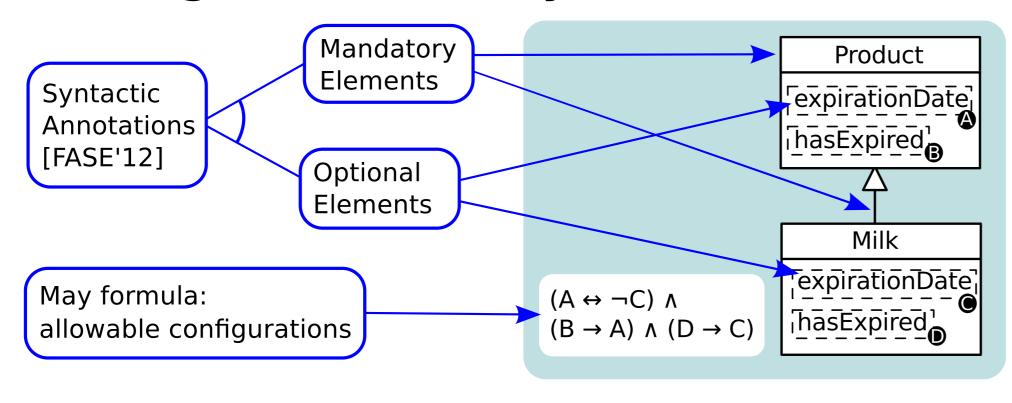


graph transformation rule

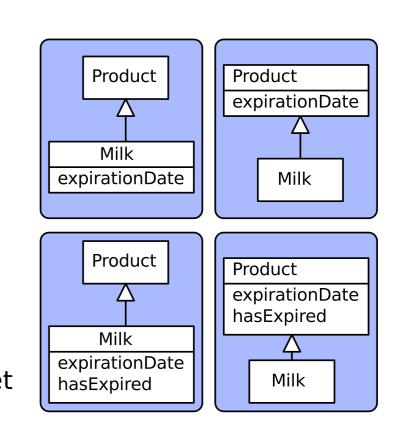
Existing model transformations

- Unambiguous model is assumed as input.
- When model contains uncertainty:
- either first remove uncertainty ...
 - premature commitment
 - reduced quality
- ... or transform all alternatives
 - hard to maintain

Explicating uncertainty with Partial Models



Semantics: set of concretizations refinement encoding Compact and exact representation of the set



Transforming Partial Models

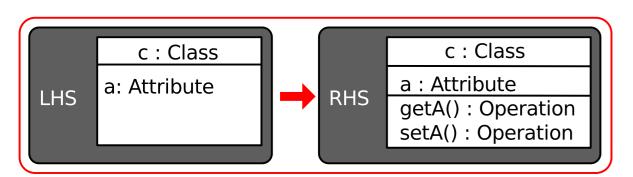
Transfer Predicates

Represent $M \stackrel{R^*}{\Rightarrow} N$ as:

$$\Phi_{M} \wedge \mathcal{R}(R,M,N) = \Phi_{N}$$

At each application point, \mathcal{R} is:

$$(\Phi_{LHS} \rightarrow \Phi_{RHS}) \wedge (\neg \Phi_{LHS} \rightarrow \Phi_{NF})$$



- $\Phi_{LHS} = c \wedge a \wedge \neg g \wedge \neg s$
- $\Phi_{RHS} = (c' \leftrightarrow c) \wedge (a' \leftrightarrow a) \wedge (g' \leftrightarrow a) \wedge (s' \leftrightarrow a)$
- $\Phi_{NF} = (\chi' \leftrightarrow \chi)$

output model input model **Product** expirationDate expirationDate hasExpired $(A \leftrightarrow \neg C) \land$ hasExpired **B** [expirationDate] $(B \rightarrow A) \land (D \rightarrow C)$ transformation rule getExpirDate() getExpirDate() hasExpired setExpirDate() setExpirDate() c : Class c : Class getHasExpired() getHasExpired() a : Attribute a: Attribute RHS |setHasExpired() |setHasExpired() [expirationDate] getA(): Operation hasExpired setA(): Operation $(A \leftrightarrow \neg C) \land$ $(B \rightarrow A) \wedge (E \leftrightarrow A) \wedge (F \leftrightarrow A) \wedge (G \leftrightarrow B) \wedge (H \leftrightarrow B) \wedge$ expressed in $(D \rightarrow C) \wedge (I \leftrightarrow C) \wedge (J \leftrightarrow C) \wedge (K \leftrightarrow D) \wedge (L \leftrightarrow D)$ as a transfer predicate propositional logic expressed as a partial model Product A Milk A (Product'⇔Product) ∧ (Milk'↔Milk)∧ Product' \(\text{Milk'} \) \(\text{\condition} \) gen_Milk_Product ∧ $(A' \leftrightarrow A) \land (B' \leftrightarrow B) \land (C' \leftrightarrow C) \land (D' \leftrightarrow D) \land (E' \leftrightarrow A) \land$ gen_Milk_Product' ^ $(A \leftrightarrow \neg C) \land (B \rightarrow A) \land$ $(F' \leftrightarrow A) \land (G' \leftrightarrow B) \land (H' \leftrightarrow B) \land (I' \leftrightarrow C) \land (J' \leftrightarrow C) \land$ $(A' \Leftrightarrow \neg C') \land (B \Rightarrow 'A') \land (E' \Leftrightarrow A') \land (B \Rightarrow 'A') \land (A \Rightarrow 'A') \land (A$ $(D \rightarrow C)$ $(\mathbb{K}' \leftrightarrow \mathbb{D}) \wedge (\mathbb{L}' \leftrightarrow \mathbb{D}) \wedge$ $(F' \leftrightarrow A') \wedge (G' \leftrightarrow B') \wedge (H' \leftrightarrow B') \wedge$ (gen_Milk_Product'⇔gen_Milk_Product) $(D' \rightarrow C') \wedge (I' \leftrightarrow C') \wedge (J' \leftrightarrow C') \wedge$ $(K' \Leftrightarrow D') \wedge (L' \Leftrightarrow D')$

Testing Correctness of Transformations

Correctness Criterion

Applying a transformation to a partial model should be the same as if we had created all its concretizations, applied the transformation to each separately, and encoded the result as a partial model.

gather all concretizations of M partial models conventional models R check equivalence using SAT apply conventional transformation rule to each R concretization individually

Conclusion

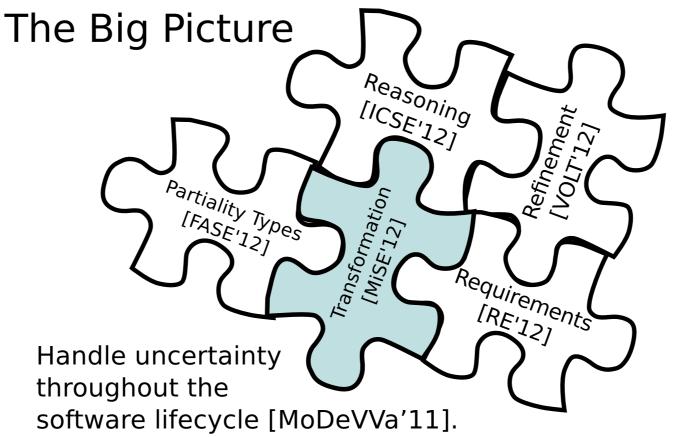
Summary

Transforming models that contain uncertainty.

- Represent uncertainty using Partial Models.
- Lift transformation rules from classical to Partial Models.
- Check Correctness Criterion for the lifted transformation .

Next Steps

- Compositionally test Correctness Criterion.
- Systematically create Transfer Predicates using FOL.
- Handle expanding/contracting model vocabularies.
- Partial Models as an Adhesive HLR Category?



References

[MoDeVVA'11.]

M. Famelis, S. Ben-David, M. Chechik and R. Salay,

Partial models: A position paper FASE'121

R. Salay, M. Famelis and M. Chechik,

Language Independent Renement using Partial Modeling [ICSE'12]

M. Famelis, R. Salay and M. Chechik,

Partial Models: Towards Modeling and Reasoning with Uncertainty [VOLT'12] R. Salay, M. Chechik and J. Gorzny,

Towards a Methodology for Verifying Partial Model Refinements

apply lifted transformation

[RE'12] Rick Salay, Marsha Chechik and Jennifer Horkoff,

Managing Requirements Uncertainty with Partial Models