Searching for Strategies that Verify MDE Toolchains

Simon Poulding, University of York & DAASE collaboration with Louis Rose, University of York
Context
Approach
Implementation
Case Study
Metamodels

In Model-Driven Engineering (MDE), model instances must comply to a metamodel that specifies attributes and associations.
Model Transformations

A common operation in MDE toolchains is the transformation of a model to another that conforms to a different metamodel.
Testing Model Transformations

To test a transformation requires a set of (possibly random) test cases in which the input data is a model instance.
Motivation

Our case study is motivated by a project in which model of behaviour is transformed into a form than can be embodied on a Lego Mindstorms robot.
Problem Statement

How can we randomly generate models for testing transformations so that a small test set of the models satisfies our testing objective?
Context-Free Grammars

A Context-Free Grammar can be used to specify how to construct ‘well-formed’ test data

\[
S \rightarrow \text{Expr} \\
\text{Expr} \rightarrow \text{Num} \mid \text{Expr Op Expr} \\
\text{Op} \rightarrow '+' \mid '-' \mid '*' \mid '/' \\
\text{Num} \rightarrow '0' \mid '1' \mid '2' \mid '3' \mid '4' \mid '5'
\]

4 + 2 * 5

3

3 * 2 - 5 / 0

/ 3 -
Stochastic Grammars

By annotating productions rule with weights, a probability distribution is defined over the language defined by the grammar

\[
\begin{align*}
S & \rightarrow \text{Expr} \\
\text{Expr} & \rightarrow \text{Num} \mid \text{Expr} \ \text{Op} \ \text{Expr} \\
\text{Op} & \rightarrow \text{`+`} \mid \text{`-`} \mid \text{`*`} \mid \text{`/`} \\
\text{Num} & \rightarrow \text{`0`} \mid \text{`1`} \mid \text{`2`} \mid \text{`3`} \mid \text{`4`} \mid \text{`5`} \\
\end{align*}
\]
Innovation: Conditional Weights

Making the weights conditional on the values of other variables introduces a limited form of context-sensitivity

\[
\begin{align*}
S & \rightarrow \text{Expr} \\
\text{Expr} & \rightarrow \text{Num} \mid \text{Expr} \text{ Op} \text{ Expr} \\
\text{Op} & \rightarrow \text{'}+\text{'} \mid \text{'}-\text{'} \mid \text{'}*\text{'} \mid \text{'}/\text{'} \\
\text{Num} & \rightarrow \text{'}0\text{'} \mid \text{'}1\text{'} \mid \text{'}2\text{'} \mid \text{'}3\text{'} \mid \text{'}4\text{'} \mid \text{'}5\text{'}
\end{align*}
\]
Innovation: Binned Scalars

Adaptively ‘binning’ scalar variables enables a compact representation of distributions over large intervals.

\[
\begin{align*}
S & \rightarrow \text{Expr} \\
\text{Expr} & \rightarrow \text{Num} \mid \text{Expr} \ \text{Op} \ \text{Expr} \\
\text{Op} & \rightarrow \ '+' \mid '-' \mid '*' \mid '/' \\
\text{Num} & \rightarrow \ '0' \mid '1' \mid '2' \mid '3' \mid '4' \mid '5' \\
\end{align*}
\]
Metaheuristic Search

To optimise a distribution, search is applied to the weights, the conditionality between variables, and the partitioning of scalar ranges

\[
S \rightarrow \text{Expr} \\
\text{Expr} \rightarrow \text{Num} \mid \text{Expr} \; \text{Op} \; \text{Expr} \\
\text{Op} \rightarrow '+' \mid '-' \mid '*' \mid '/' \\
\text{Num} \rightarrow '[0,228]' \mid '[229,433]' \mid '[434,511]' 
\]
Context

Approach

Implementation

Case Study
HUTN

HUTN is a textual notation for model instances

```
A {
  id: “F45”
  b: B { cost: 7}
  c: C { size: 25}, C {size: 30}
}
```
Grammar To Emit HUTN

We use a stochastic context-free grammar that emits HUTN which complies with the chosen metamodel.

HUTN Grammar

\[
S \rightarrow A \\
A \rightarrow 'A' \{ 'id:' String 'b:' B1..* 'c:' C0..* '}' \\
B1..* \rightarrow B | B , ' B1..* \\
B \rightarrow '{' 'cost:' Cost '}' \\
Cost \rightarrow ['0,100]' \\
\ldots \rightarrow \ldots
\]
Optimisation Process

The HUTN grammar is optimised by evaluating set of models sampled from the candidate grammar.

1. Transform metamodel to HUTN grammar
2. Sample instances from stochastic grammar
3. Transform HUTN instances to model instances
4. Measure fitness of model instances
5. Use fitness to optimise grammar
Physical Implementation

Search executable optimises and samples from the grammar; in a servlet, instrumented MDE transformation converts HUTN to instances and assesses; components communicate over HTTP
Source Metamodel

Specifies the models that are the inputs to the transformation under test

http://lego.wikia.com/wiki/File:LEGO_Mindstorms_NXT.jpg.jpeg
Objective: Statistical Testing

Want to maximise frequency of covering every rule, guard, and condition as frequently as possible in order to minimise test set size
Experiments

Compare efficiency of optimised and unoptimised grammar; random search as measure of ‘difficulty’

- Optimised using hill-climbing (800 evaluations)
- Unoptimised (‘uniform’ distribution)
- Optimised using random search (800 evaluations)
Results

Number of test cases to cover all elements (with a 90% likelihood) - smaller is better

- Optimised (hill-climb)
- Unoptimised
- Optimised (random search)

Number of test cases
Other Outcomes
Process highlighted ambiguities and missing information in the original metamodel
Future Work

Automate metamodel to HUTN grammar conversion; speed up evaluation
Further Details

Simon Poulding, Robert Alexander, John A. Clark, and Mark J. Hadley
*The Optimisation of Stochastic Grammars to Enable Cost-Effective Probabilistic Structural Testing*
Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2013)
(to appear)

Louis M. Rose and Simon Poulding
*Efficient Probabilistic Testing of Model Transformations using Search*
Proceedings of 1st International Workshop on Combining Modelling and Search-Based Software Engineering (CMBSE 2013)
(to appear)