A Compositional Approach to Bidirectional Model Transformation

ICSE’09 New Ideas and Emerging Results

Soichiro HIDAKA Zhenjiang HU Hiroyuki KATO
National Institute of Informatics, Japan

Keisuke NAKANO
The University of Electro-Communications, Japan

Introduction

In bidirectional model transformation, modifications propagate from source models to target models as well as from target to source. Although bidirectional model transformation plays an important role in model-driven software development, lack of clear semantics of composition is one of open problems.

Proposed Approach and Results

Compositional graph transformation language UnQL is extended for bidirectional model transformation by
- Editing primitives (replace, delete, extend) [1]
- Bidirectional interpretation of each graph constructors and combinator [2]

Models as Edge-labeled Graphs

Models are internally represented by edge-labeled graphs.

Class diagram: an example of models to be transformed

Model Transformations in UnQL+

Transformation to prefix every name of the class by “class_” can be expressed in UnQL+ as

```
replace (\$Name : \{} by \{'class_'.\$Name:\{}\} where \\
\{\_.,\_Class\.name\_String\: (\$Name: \{}\} in \$classDB
```

Bidirectional Evaluator and its property

Every UnQL+ program is translated [1] into UnCAL in which fixed number of constructors and combinators are combined to form a bigger transformation.

Formal semantics: union (∪) example

Two transformations are executed componentwise and combined.

\[
E ::= \{} | \{ L : E \} | E \cup E
\]

```
\begin{align*}
E & ::= \{} | \{ L : E \} | E \cup E \\
& \quad | \& x := E | \& y | () | E \oplus E | E @ E
\end{align*}
```

```
\begin{align*}
E & ::= \{} | \{ L : E \} | E \cup E \\
& \quad | \& x := E | \& y | () | E \oplus E | E @ E \\
& \quad | \& \lambda (LabelVar, Var).E(Var)
\end{align*}
```

Bidirectionalization

Syntax of UnCAL graph algebra

```
E ::= \{} | \{ L : E \} | E \cup E
```

```
\begin{align*}
E & ::= \{} | \{ L : E \} | E \cup E \\
& \quad | \& x := E | \& y | () | E \oplus E | E @ E
\end{align*}
```

```
\begin{align*}
& \quad | \& \lambda (LabelVar, Var).E(Var)
\end{align*}
```

Well behavedness

No change on the target $g$ should give no change on the source (environment) $\rho$.

\[
\frac{\rho \in g}{\rho \in g}
\]

Another forward transformation from the modified source $\rho'$ produces $g'$ again.

\[
\frac{\rho' \in g'}{\rho' \in g'}
\]

Impact and Future Work

- Demonstrate that functional approach is helpful to give bidirectional semantics in a formal and concise way
- Demonstration available at http://www.biglab.org/
