

# Reimplementation of the OCL/DL Translation

06.06.06

Denis Lohner

5<sup>th</sup> International KeY Symposium  
Speyer

# The KeYOclParser (1)

- Input:
  - one single OCL-expression (invariant, pre- or postcondition) represented by a String
  - Context of the expression
- Output:
  - Term representing the given expression
- Assumption:
  - Input is correct OCL (OCL standard)

# The KeYOclParser (2)

- Tried to be very close to the semantics described in the current OCL-specification standard
  - Associativity of operators (And/Or are both left-associative with same precedence)
    - $a \text{ or } b \text{ and } c = (a \text{ or } b) \text{ and } c$

# What is supported (1)

## (operators, lowest precedence first)

- Logical Expressions
  - implies, and, or, xor
- Relational Expressions
  - =, <>, <, >, <=, >=
- Additive/Multiplicative Expressions
  - +, -, \*, /
- Unary Expressions
  - not, - (sign)

# What is supported (2)

- Postfix Expressions
  - Property Calls
  - @pre
- Primary Expressions
  - true, false, null
  - Literals (Numbers)
  - Literal Collections
  - If-Then-Else Expression
  - Parentheses

# What is not supported (yet)

- String/Enumeration literals
- Reals
- Some property calls (e.g. `→iterate`)
  - OCL standard library is huge!
- Collections of collections
  - **Only exception:** Navigation over multiple associations with multiplicity greater than 1
- Let Expressions (not yet – will be next step)

# What are Property Calls (1)

- Associations
  - Multiplicity  $\leq 1$ , then translated to objects
  - Translated to collections otherwise
- Attributes / Methods
  - Translated to terms
- Method Parameters
  - Translated to terms (only pre- and post-conditions)

# What are Property Calls (2)

- Typenames
  - followed by `allInstances()`
  - In expressions like `oclIsKindOf`
- Property Calls defined by the OCL Standard Library (Property Calls on collections)
  - `forAll`, `exists`, `isUnique`, `collect`,  
...
- Parser is very strict concerning case sensitivity and (empty) parentheses

# Supported Property Calls (1)

- on collections:
  - forAll, exists
  - select, reject
  - collect
  - includes, excludes
  - isEmpty, notEmpty
  - isUnique
  - size, sum, count
- on objects:
  - oclIsKindOf, oclIsTypeOf, oclAsType
- Array access (not OCL-standard)
  - get

# Supported Property Calls (2)

- Not supported yet:
  - asSet, asBag, asSequence
  - includesAll, excludesAll
  - Collection-specific property calls (e.g. union)
- Not supported (and won't be supported in the near future)
  - collectNested, flatten, product
  - iterate

# Translation Details (Predicative / Functional)

- Collections can be described in two ways (see Diploma Thesis of Uwe Keller)
- Functional Description:
  - Collection and associations are described by **functions**
  - Functions are defined by collection-specific axioms
  - Sort of the collection-function is a collection-sort

$Class1.allInstances() \rightarrow size()$  will be represented by:

$size(allInstances_{Class1})$  where  $allInstances_{Class1} : \rightarrow Set_{Class1}$  with axiom:

$\forall Class1\ x ; (x.<created> \wedge x \neq null) \leftrightarrow includes_{Class1}(allInstances_{Class1}, x)$

# Translation Details (Predicative / Functional)

- Predicative Description:
  - Collection represented by a **formula**, associations by **predicates**
  - Formula can be interpreted as a restriction for elements in this collection
  - Quantified over a variable representing one element of the collection

`Class1.allInstances()->select(x | x.balance=0)->forall(x | x.isInvalid)`

will be represented by:

$\forall \textit{Class1 } x ; x.\textit{balance} = 0 \rightarrow x.\textit{isInvalid}$

# Translation Details (Predicative / Functional)

- Which description will be used?
  - Mainly predicative, since KeY can handle this more easy
  - But: Not applicable for all property calls
  - Conclusion: Depending on last property call, the description is chosen
    - Functional for: size, sum, count

# Don't be surprised

- At the moment, all created axioms are included in the resulting term
- If there is a (spelling-)mistake in the ocl-expression, it is likely you get an exception since the parser assumes correct input
  - Misspelled property calls (case sensitivity!)

Thank you for your attention!

Any questions?