Patterns in OCL

Burkhart Wolff

Université Paris-Sud
Pattern-Matching Lambdas

• Proposal:
  • Hidden second-order combinators, implicitly accepting a lambda buried under first-order notation, are:

    ->iterate  ->exists  ->forall  ->select
    ->collect  ->any    ->isUnique.

S->select(PATTERN | P x)

for example:

    S->select(Seq{__, 3, a, ...} | a >= 15)

or

    S->select(Tuple{name='mueller', sex=male, age= x, ...} | x >= 21)
Pattern-Matching Lambdas

• Proposal:
  • Hidden second-order combinators, implicitly accepting a lambda buried under first-order notation, are:

for example

\[ S \rightarrow \text{select}(\text{Seq}\{\_\_\_\_3, a, \ldots\} \mid a \geq 15) \]

or

\[ S \rightarrow \text{select}(\text{Tuple}\{\text{name}='\text{mueller}', \text{sex}=\text{male}, \text{age}=x, \ldots\} \mid x \geq 21) \]

or

\[ S \rightarrow \text{select}(a \text{ in Employee} \mid P \ a \ ) \quad \text{for} \quad (S \rightarrow \text{select}(a \mid a.\text{oclIsKindOf(Employee)} \text{ and } P \ a \ )) \]
Pattern-Matching Lambdas

Proposal:
- Hidden second-order combinators, implicitly accepting a lambda buried under first-order notation, are:

Possibility: implicit Tuple-notation for Classes:
```plaintext
class Employee is Person
    + salary : Integer[0..1]
    + department_id : Integer [1]
end
```

Example:
- $S \rightarrow \text{select}(\text{Employee}\{\text{salary} = x, \text{department_id}=5, \ldots \} \mid x \neq \text{null and } x>2000)$