Exercise: OCL

Purpose

Let you practise reading and writing OCL constraints.

Here are a couple more useful OCL operations on collections that were not explained in the slides. (There are more: for full details, see section 11.7 of the OCL spec.)

Suppose \( c \) is a Collection of elements of type \( T \), and \( t : T \). Then we can write:

- \( c->\text{includes}(t) \)
  a Boolean expression that will be true iff the element \( t \) is equal to an element of the collection (exercise: write this in terms of \( \exists \) instead: yet another example of the non-parsimony of the UML/OCL language!)

- \( c->\text{including}(t) \)
  an expression that evaluates to a collection which is the same as \( c \) except that \( c \) has been added to the collection. (If \( c \) is a sequence, \( t \) is added as the last element of the new collection; if it is a bag or a set, the obvious thing happens.)

These questions refer to the following diagram extracted from the OCL specification.
1 Question 1

Translate into English:

1. In the context of a Person:
   
isMarried implies age > 15

2. context Company inv:
   
   numberOfEmployees = employee->size()

3. context Person::income(d:Date) : Integer
   
   pre: d.laterThan(self.birthDate)
   post: if age < 18
       then result < 100
       else result < 200
   endif

4. In the context of bigBank : Bank:

   bigBank.customer -> collect(p : Person | p.managedcompanies)
   -> asSet() -> size() >= 3

What is the difference between this and

bigBank.customer -> collect(p : Person | p.managedcompanies)
-> size() >= 3

?
2 Question 2

Translate into OCL:

1. The length of a person’s first name is always less than 20 characters, and so is the length of their last name.

2. Anyone who manages a company is an employee of that company. (You could write this in context Person – making it an invariant of Person – or in context Company – making it an invariant of Company. Try it both ways.)

3. Every company has a male employee.

4. It is a class invariant of Person that nobody can have more than 5 bank accounts.

5. Nobody can have two employments with companies that have identical names.