

OCL: Object Constraint Language

Prof. Daniel Riesco

Introducción

- Lenguaje formal para expresar restricciones libres de efectos colaterales.
- No es un lenguaje de prog. (no hay flujo de control).
- No se puede invocar procesos.
- Tiene tipos.
- Propósitos:
 - especificar invariantes
 - especificar pre y post condiciones en los métodos
 - como lenguajes de navegación
 - especificar restricciones en las operaciones
 - operación(a, b, c) = expresión

Componentes

- Self
- Invariantes (<<invariant>>)
 nbreTipo :: nbreOp(p1:T1, ...) : tipoRetorno
 pre: p1 ...
 post: result = ...
- Expressions
- Tipos básicos:
 - Boolean: and, or, not, implies, if-then-else
 - Integer: *, +, /, abs, mod, min, max, ...
 - Real: *, +, /, floor, <, ...
 - String: toUpper, concat, size, substring(i:integer, s:integer),...
- Comentarios --

Propiedades

- un atributo
- un extremo de una asociación (AssociationEnd)
- un método con *isQuery* en true
 - unTipo
 - self.propiedad
- AssociationEnd
 - cardinalidad 1 -> retorna un objeto o un set de un elem.
 - cardinalidad n -> retorna un set
 - cardinalidad n {ordered} -> retorna una sequence
- Ejemplos
 - Las personas casadas son mayores de edad
 - Una compañía tiene al menos 50 empleados
 - un casamiento es entre hombres y mujeres
 - Una persona no puede tener una esposa y un esposo

Propiedades

- Clase asociación
 - se usa el punto y el nombre de la asociación
Person self.job
 - desde la clase asociación hacia los objetos que participan
Job self.empleados
- Predefinidos:
 - oclType: OclType
 - oclIsTypeOf(t: OclType): Boolean
 - oclIsKindOf(t:OclType) :Boolean, =, <>
- Person.allInstance->forall(p1, p2 | p1<> p2 ...}
- Colecciones
 - Set {1 .. 5, 7}
 - Sequence {1, 3, 45, 2}
 - Bag {1, 1, 4}

Collection

- Collection es subtipo de OclAny
- Set, Bag, Sequence son subtipos de Collection
- Operaciones:
 - collection->select(v : type | boolean-expr-with-v)
 - collection->reject(v : type | boolean-expr-with-v)
 - collection->collect(v : type | expr-with-v)
 - collection->forall(v : type | boolean-expr-with-v)
 - collection->exists(v : type | boolean-expr-with-v)
- Otras operaciones son: size, count, isEmpty, notEmpty, sum, includes (obj pasado como parámetro pertenece a la colección), includesAll (inclusión de colección),

Collection

- Cast
 - $t1 \rightarrow t2$ cdo son = o $t1$ es subtipo de $t2$
 - $\text{Collection}(t1) \rightarrow \text{Collection}(t2)$ cdo $t1$ es subtipo de $t2$
 - es transitivo.
- Valor previo en postcondición: nbreAtrib@pre
- OclType
 - `type.name` : String
 - `type.attributes` : Set(String)
 - `type.associationEnds` : Set(String)
 - `type.operations` : Set(String)
 - `type.supertype` : Set(OclType)
 - `type.allSuperTypes` : Set(OclType)
 - `type.allInstance` : Set(type)

Set

- `set->union(set2 : Set(T)) : Set(T)`
- `set->union(bag : Bag(T)) : Bag(T)`
- `set->intersection` (con set and bag)
- `set->including(obj : T) : Set(T)` (Agrega obj a set)
- `set->excluding(obj : T) : Set(T)` (Elimina obj a set)
- `set->symmetricDifference(set2 : Set(T)) : Set(T)`
- `set->asSequence : Sequence(T)` (orden aleatorio)
- `set->asBag : Bag(T)`
- `=, -, select, reject, collect`

Bags and Sequences

- Operaciones de una bolsa (Bag)
 - =, union, intersection, including, excluding, select, reject, collect, count, asSequence, asSet
- Operaciones de una Secuencia (Sequence)
 - count, =, union (es una concatenación), append (agrega el obj al final), prepend (agrega obj al principio), subSequence(lower:integer, upper:Integer), at(i :Integer), first, last, excluding (quita todas las ocurrencias de obj), select, reject, collect, iterate, asBag, asSet

Ejemplos del Metamodelo

- Asociación
 - Los finales de asociación deben tener un nombre único dentro del final de Asociación.
 - A lo sumo un final de asociación puede ser una agregación o composición.
- Final de Asociación
 - El clasificador de un final de asociación no puede ser un DataType ni una Interface.
 - Una instancia es de composición no puede pertenecer a más de una instancia compuesta.
- Clasificador
 - No puede haber dos métodos con la misma signatura.
 - No puede haber dos atributos con el mismo nombre