**Unchangeable Graph Nodes**

*System Model*

The reachability analysis has two inputs: a set of initial graph nodes and a set of story diagrams that model how the initial graph nodes may be modified.

*General Idea*

The set of initial graph nodes may contain nodes that are needed for the reachability analysis, but they are not modified by the given set of story diagrams. These nodes, however, are part of every graph which is reachable during the analysis. We call these nodes *unchangeable graph nodes*. All nodes which are actually modified by the given set of story diagrams are called *changeable graph nodes*. For improving the efficiency of the analysis, the two sets of nodes should be maintained separately.

The unchangeable graph nodes are kept once per reachability graph while the changeable nodes are contained in every single state of the reachability graph. Then, all states of the reachability graph share the same set of unchangeable nodes.

*Conditions*

The conditions for unchangeable nodes are based on the set of story diagrams as well as the underlying Ecore meta-model over which the nodes in the story diagrams are typed. Since we can’t state which concrete objects of the initial graph objects are unchangeable, we identify the EClasses where all instances are unchangeable nodes. I.e., given an object a which is an instance of class A, either all objects of class A are unchangeable or none.

To ease the definition of unchangeable nodes, we first state the conditions that cause objects of a class A to be considered changeable.

1. There exists a story diagram which contains an object variable of type A which has binding operator CREATE or DESTROY.
2. There exists a story diagram which contains an object variable of type of A which has an AttributeAssignment.
3. There exists a story diagram which contains a link variable originating from an object of class A which has binding operator CREATE or DESTROY.
4. There exists a story diagram which contains a link variable pointing to an object of class A which has binding operator CREATE or DESTROY and which is bidirectional.
5. There exists a class B which is changeable and (recursively) contains objects of class A.
6. There exists a class B which is changeable and objects of class A recursively contain objects of class B (creating or deleting B changes the reference of A to B)
7. There exists a class B which is changeable and A is a subclass of B (recursively), (superclasses of B or not necessarily changeable).

We define the following conditions for a node of class C to be considered unchangeable.

1. A node of class C may only be unchangeable if it is contained in the initial graph nodes.
2. A node of class C may only be unchangeable if C is not considered to be changeable by the above conditions for changeable nodes.