

The GrGen.NET ToDo-List

Frontend (Java):

ToDo:

- In replace/modify-part `typeof` does not work when used with retyping of nodes and/or edges.
- For .grg-files, that are empty except for the `actions` statement, GrGen wrongly reports an error.
- In declarations of enumeration types (keyword `enum`) user defined integer values can be assigned to elements. However, on the RHS of such „assignments“ it should be possible to use already defined elements of that enumeration types in expressions (e.g., `... x = 42, y = x + 3...`). However, GrGen does not accept this.
- GrGen crashes with a `NullPointerException` if the .grg file specifies a non-existing model file.
- GrGen crashes when a newly created edge is returned, saying that "the element 'type3', that is neither a parameter, nor contained in LHS, nor in RHS, occurs in a return". Crashes in `...ast.RuleDeclNode.checkReturnedElemsNotDeleted`. (should_pass/ret_003_fe.grg)
- Where warnings should be raised:
 - In modify parts, if a graph element occurs *inside* as well as *outside* a `delete` statement.
 - If an assignment `x.a = ...` occurs inside an `eval`, but the graph element `x` will be deleted on a rewrite.
 - If one or more attributes of a newly created node are not initialized by the eval part of the respective rule.
 - If the types of potentially homomorphic pattern nodes have no common subtype (because non-injective matching is *impossible* in this case).
- The statements `actions` and `model` should be removed from the specification language of GrGen.NET. However, if such a statement occurs at the beginning of a specification, a deprecated-warning should be raised. The `using` keyword will be kept, of course.
- In some error messages appear corrupt coordinates (this is where a „?“ appears instread of line and column).

Done:

- In modify-parts the error detection wrongly reports invalid reuse of nodes and edges.
{seems to work now—Batz 7/17/2007}
- Dangling edge graphlets on the RHS should work, if the edge is a reused one and if all incident pattern nodes of that edge are also reused. However, if such an edge is retyped, GrGen wrongly reports an error.
{seems to work now—Batz 7/17/2007}
- Annotations of anonymous nodes and edges do not work.
{implemented—Batz 7/14/2007}
- If the filename of a .grg-file does not conform with the name given along with the **actions** keyword, an error is raised (which is just the right behaviour). However, the output file is generated all the same, which should not happen.
{fixed—Batz 7/13/2007}
- Error detection for the **return** statement is erroneous.
{seems to work now—Batz 7/13/2007}
- At test case should_fail/ret_001.grg: The signature of the rule demands a type **AB**. However, if you return a type **C** (that is no subtype of **AB**) no error is reported, which were the right behaviour.
{as the return stuff now works, this works, too—Batz 7/13/2007}

C#-Searchplan-Backend (Java):

ToDo:

- The C#-code generated for /should_pass/basic_027.grg is not correct. Because of "x -e-> y;" in the modify part, "delete(x);" is ignored.

Done:

- Float constants are not correctly emitted.
{fixed—Kroll 2007-07-14}

Backend (C#):

ToDo: Done:

Other things like, e.g., bugs of unknown origin

ToDo: Done: