The GrGen.NET ToDo-List

Frontend (Java):

ToDo:

- Large graphs in GrGen rules cause very long running times. Maybe there are memory leaks and/or sloppy coding. Use a profiler to find out where and what.
- Implementation of the `RETURN` child of class `GraphNode` does not follow the presumably intended design principles of the AST implementation: Identifiers are not resolved to instances of `NodeDeclNode` or `EdgeDeclNode`. However, if we fix this, this will have consequences for the check phase, where several things regarding the `return` statement are checked.
- Invent attribute type `object` that only supports the operators `==` and `!=`.
- Check the cause, why some test cases still do not work as required. Dont forget to check should_fail for cases that do *crash* instead of reporting an error.
- Improve error messages: Some of them are errorneous and some are even completely malformed. Possible reason: implemented too generic. Possible Location in Code: Resolvers. Possible solution: Implement some more resolvers (but only for error reporting reasons).
- `test` rules don't work yet.
- Where warnings should be raised:
  - 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).
- In some error messages appear corrupt coordinates (this is where a „?“ appears instread of line and column).

Done:

- For empty .grg-files and for .grg-files that are empty except for a `using` statement GrGen wrongly reports an error.
  {fixed—Batz 1. Aug. 2007}

- Where warnings should be raised:
  - In modify parts, if a graph element occurs *inside* as well as *outside* a `delete` statement.
    {implemented—Batz 6. Aug. 2007}
  - On the RHS, if for a returned element homomorphic matching is allowed with a deleted node (hom-delete-return conflict).
    {implemented—Batz 1. Aug. 2007}
- GrGen crashes with a NullPointerException if the .grg file specifies a non-existing model file.
  {fixed—Batz 30. Jul. 2007}
- Implement enums correctly. This includes implicit type casts to `int` as well as use of already defined items in an enum delcaration as well as use of foreign enum items if they are *fully* qualified. If they are *not* qualified an error must be reported. The def-before-use law must also hold if items from other enum types are used in the definition of an enum item.
  {Seems to work now—Batz 30. Jul. 2007}
- 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.
  {I did it—Batz 25. Jul. 2007}
- 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.
  {works—Batz 24. Jul. 2007}
- Bug: If an error is reported in a model (.gm) file, the execution of the frontend is not aborted before building the IR.
  {fixed—Batz 24. Jul. 2007}
- In replace/modify-part `typeof` does not work when used with retyping of nodes and/or edges.
  {seems to work now—Batz 18. Jul. 2007}
- GrGen crashes when a newly created edge is returned, saying that "the element 'type3', that isneither a parameter, nor contained in LHS, nor in RHS, occurs in a return". Crashes in ...ast.RuleDeclNode.checkReturnedElemsNotDeleted.

(should_pass/ret_003_fe.grg)
{fixed—Batz 17. Jul. 2007}

- In modify-parts the error detection wrongly reports invalid reuse of nodes and edges.
  {seems to work now—Batz 17. Jul. 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 17. Jul. 2007}

- Annotions of anonymous nodes and edges do not work.
  {implemented—Batz 14. Jul. 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 genrated all the same, which should not happen.
  {fixed—Batz 13. Jul. 2007}

- Error detection for the `return` statement is errorneous.
  {seems to work now—Batz 13. Jul. 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 13. Jul. 2007}

C#-Searchplan-Backend (Java):

ToDo:

- Check wether the code generated for access to enum type attributes and constants is correct.

- 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:

- Enum constants are printed as integers in enum expressions.
  {fixed—Kroll 19. Jul. 2007}

- Attributes with names of reserved keywords should be prefixed by an '@'.
  {fixed—Kroll 19. Jul. 2007}

- Fixed casts to strings.
  {fixed—Kroll 19. Jul. 2007}
- Fixed conditions containing casts.
  {fixed—Kroll 19. Jul. 2007}
- Float constants are not correctly emitted.
  {fixed—Kroll 17. Jul. 2007}

Backend (C#):

ToDo: Done:

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

ToDo: Done: