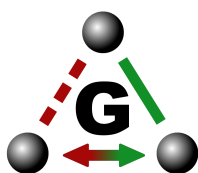


GGL Frequently Asked Questions



Christoph Flamm^{1,1} and Martin Mann²

¹ Institute for Theoretical Chemistry, Vienna University



² Bioinformatics Group, University of Freiburg



<http://www.tbi.univie.ac.at/software/GGL/>

Version September 17, 2014

¹Send comments to xtof@tbi.univie.ac.at or
mmann@informatik.uni-freiburg.de

Built for GGL version 4.1.0

Contents

1	General Questions	1
1.1	Questions by Hans Peschke to be distributed when answered .	1
1.2	Graph Grammars and Graph Rewrite Systems	2
1.2.1	What are Graph Grammars?	2
1.2.2	What are Graph Rewrite Systems?	2
1.2.3	How does a Double Push Out (DPO) Approach work? .	2
1.2.4	What rewrite approach is used within the GGL?	2
1.3	Graph Definition	2
1.3.1	What is GML?	2
1.3.2	How to specify a graph in GML?	2
1.4	Graph Grammar Rule Definition	2
1.4.1	What is a graph grammar rule?	2
1.4.2	How to specify a graph grammar rule in GML?	2
1.4.3	Are wildcards allowed?	3
1.4.4	Can I further constrain the matching?	3
2	GGL C++ Questions	4
2.1	General	4
2.1.1	How to compile the GGL library?	4
2.1.2	How to use and link the GGL library?	4
2.1.3	Where to find examples?	4
2.2	Graphs	5
2.2.1	What graph object is used?	5
2.2.2	What types of graphs are possible?	5
2.2.3	How to access graph properties?	5
2.2.4	How to parse a graph from GML definition?	5
2.2.5	How to write a graph in GML format?	5
2.2.6	How to write a graph in GXL format?	5
2.3	(Sub)graph Matching	5
2.3.1	What is the <code>sgm::Graph_Interface</code> and what is it used for?	5

2.3.2	How to search for a subgraph within a target graph? .	5
2.3.3	How to find graph isomorphisms / do graph matching?	6
2.3.4	What matching algorithms are used and available? . .	6
2.3.5	What is the VF2 algorithm?	6
2.3.6	How are matches handled?	6
2.3.7	Is there a possibility for automorphism/symmetry exclusion?	6
2.4	Graph Grammar Rules	6
2.4.1	How to read/load a graph grammar rule from GML format?	6
2.4.2	How to apply a graph grammar rule?	6
2.4.3	How to handle/apply multi-component graph grammar rules?	6

Chapter 1

General Questions

1.1 Questions by Hans Peschke to be distributed when answered

- Welche Nomenklatur von Graph Grammars benutzen Sie? (d.h. ein Verweis in die Literatur zur Theorie der Graph Grammars, sowie deren Implementierung von wesentlichen Algorithmen, waere sehr hilfreich, um sich in Verbindung mit ihrer Library tiefer in das Thema einzulesen)
- Wie erzeugt man Graphen?
- Ob und wie kann man Graphen durch die Anwendung von Produktionsregeln erzeugen automatisch erzeugen lassen?
- Welche Schritte sind notwendig, um Graphen nach gewissen Regeln zu veraendern?
- Ideen fuer Beispiele:
 - Start mit einem Knoten und rekursive Ersetzung der Knoten durch kleine Graphen
 - Binaerbaum
 - Erzeugung eines k-dim. cubes
 - vollstaendiger Graph

1.2 Graph Grammars and Graph Rewrite Systems

1.2.1 What are Graph Grammars?

1.2.2 What are Graph Rewrite Systems?

1.2.3 How does a Double Push Out (DPO) Approach work?

1.2.4 What rewrite approach is used within the **GGL**?

1.3 Graph Definition

1.3.1 What is **GML**?

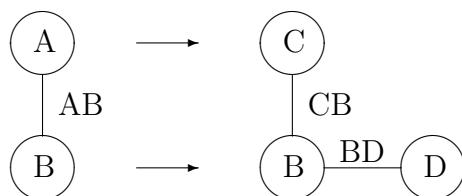
1.3.2 How to specify a graph in **GML**?

1.4 Graph Grammar Rule Definition

1.4.1 What is a graph grammar rule?

1.4.2 How to specify a graph grammar rule in **GML**?

Consider the following graph grammar rule that adds a new node “D” and a connecting edge to “B” with label “BD”, relabels a node “A” to “C” and relabels the connecting edge too “B” from “AB” to “CB”.



An according GML encoding would look like this:

```

rule [
  ruleID "Relabel node 1 and edge 1-2 and add node 3"
  left [
    node [ id 1 label "A" ]
    edge [ source 1 target 2 label "AB" ]
  ]
  context [
    node [ id 2 label "B" ]
  ]
  right [
    node [ id 1 label "C" ]
    node [ id 3 label "D" ]
    edge [ source 1 target 2 label "CB" ]
    edge [ source 2 target 3 label "BD" ]
  ]
]

```

Note: A standard DPO approach would remove node “A” and add a new node “C” (including subsequent rewiring) instead of doing a direct node relabeling. Within the GGL framework, we enable the direct relabeling of nodes that make the following “rewiring” of the loose ends (cause by the removal of “A”) with the newly introduced node “C” obsolete. But this *relabeling is exclusive to nodes!* The “relabeling” of edges is done via the removal of edge “AB” and the adding of a new “CB” edge.

1.4.3 Are wildcards allowed?

1.4.4 Can I further constrain the matching?

Chapter 2

GGL C++ Questions

2.1 General

2.1.1 How to compile the GGL library?

2.1.2 How to use and link the GGL library?

```
#####  
# the GGL installation directory (after 'make install')  
GGL_install_dir=....  
  
# ggl specific compilation flags  
GGL_CXXFLAGS = -I$(GGL_install_dir)/include  
  
# ggl linker flags (note: the order is important for g++)  
# if you need the whole package  
GGL_LDFLAGS = -L$(GGL_install_dir)/lib -lgglchem -lnspdk -lsqd -lggl  
  
# if you need the ggl package only  
GGL_LDFLAGS = -L$(GGL_install_dir)/lib -lggl -lsgm_vf2 -lsgm -lvf2  
#####
```

2.1.3 Where to find examples?

within tests

2.2 Graphs

2.2.1 What graph object is used?

```
#include <ggl/Graph.hh> : ggl::Graph == boost::adjacency_list
```

2.2.2 What types of graphs are possible?

only labeled undirected graphs

2.2.3 How to access graph properties?

either via boost functions or via `ggl::Graph_boost` i.e. `sgm::Graph_Interface` (link)

2.2.4 How to parse a graph from **GML** definition?

example of `ggl::Graph_GMLparser`

```
1  \caption{example of a graph creation from \GML{}} string}
2  ggl/Graph.hh : ggl::Graph == boost::adjacency_list
```

2.2.5 How to write a graph in **GML** format?

example of `ggl::Graph_GML_writer`

2.2.6 How to write a graph in **GXL** format?

example of `ggl::Graph_GML_writer`

2.3 (Sub)graph Matching

2.3.1 What is the `sgm::Graph_Interface` and what is it used for?

2.3.2 How to search for a subgraph within a target graph?

example of `sgm::SubGraphMatching` using `sgm::SGM_VF2`

2.3.3 How to find graph isomorphisms / do graph matching?

example of `sgm::GraphMatching` using `sgm::GM_VF2`

2.3.4 What matching algorithms are used and available?

discuss possible integration of any algorithm and link to VF2 algorithm as current standard approach

2.3.5 What is the VF2 algorithm?

reference and discuss VF2 algorithm

2.3.6 How are matches handled?

discuss `sgm::Match_Reporter` interface and show different examples for counting, output writing etc.

2.3.7 Is there a possibility for automorphism/symmetry exclusion?

2.4 Graph Grammar Rules

2.4.1 How to read/load a graph grammar rule from GML format?

example `ggl::Rule_GMLparser`

2.4.2 How to apply a graph grammar rule?

discuss usage of matching framework (link) example `ggl::MR_ApplyRule` with `ggl::GS_STL` storage

2.4.3 How to handle/apply multi-component graph grammar rules?

discuss recursive component-wise rule application as done within `toyChemUtil`