

# pyGK User Manual

Marc Provost  
McGill University

December 18, 2003

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Graph Representation</b>	<b>3</b>
<b>3</b>	<b>Graph Manager</b>	<b>3</b>
3.1	AGLGenerator . . . . .	3
3.2	Example . . . . .	4
<b>4</b>	<b>Graph Rewriting System</b>	<b>4</b>

# 1 Introduction

pyGK, for “python graph kernel” is a set of classes supporting a representation for higraphs.

## 2 Graph Representation

## 3 Graph Manager

### 3.1 AGLGenerator

The AGLGenerator module generates a xml-based representation for graphs. Given a graph, it will produce xml following the *agl* specification, which can be located in the file *agl.dtd*<sup>1</sup>. The agl language, which is an acronym for “a graph language”, was kept as simple as possible. One tag is found for each GraphElement:

- Node (e.g. `<node id="", type="" />`)
- Graph (e.g. `<graph id="", type="" />`)
- Int (e.g. `<int id="", type="Int", value="0" />`)
- String (e.g. `<string id="", type="String", value="" />`)
- Bool (e.g. `<bool id="", type="Bool", value="" />`)
- Float (e.g. `<float id="", type="Float", value="" />`)
- List (e.g. `<list id="" type="List">`)
- SymbolTable (e.g. `<symb id="" type="SymbolTable">`)

In agl, the representation for each inner graph is stored in a separate file. This is done in order to be able to load any inner graph in multiple contexts. The knowledge of the location of the xml representation of a given graph is external to the loader. Thus, it must be specified when AGLLoader is instantiated via a python dictionary, mapping each graph’s globalID to its file.

---

<sup>1</sup>XMLTools/dtd/agl.dtd

### 3.2 Example

## 4 Graph Rewriting System