Manipulate a DSL

DOT

Ingénierie Des Modèles

University of Rennes 1 - ISTIC
Master 2 Génie Logiciel - Apprentissage

UNIVERSITÉ DE RENNES 1

Mickaël CALLIMOUTOU

1. Domain Specific Language : DOT

Source: https://en.wikipedia.org/wiki/DOT_%28graph_description_language%29

Dot is domain specific language.
DOT is a plain text graph description language. It is a simple way of describing graphs that both humans and computer programs can use. DOT graphs are typically files that end with the .gv or .dot extension

2. External DSL : DOT

The following content was written in a **.dot file** ***(output/model.dot)***, using a simple text editor:

```
graph model {
        top;
        v1;
        v2;
        v3;
        v4;
        v5;
        bottom;
        top -> v1 -> v2;
        top -> v3;
        top -> v4;
        v2 -> bottom;
        v3 -> v5;
        v4 -> v5;
        v5 -> bottom;
}
```

This file describe a simple graph with 7 vertices and 8 edges.

3. Internal Java DSL : JGraph

In order to describe graphs in Java, I decided to use **JGraph** library.
This library allow you to describe a graph using java syntax, and then export the result in DOT syntax ***(output/generateWithJava.dot)***, thanks to DOTExporter object.

The following content was written in a **.java file**, using **eclipse** as IDE:

```
public class Main {

/** Main method.
 *  Write a dot file describing a simple graph */
public static void main(String args[]) throws IOException {
      UndirectedGraph<String, DefaultEdge> stringGraph = createStringGraph();
      DOTExporter<String, DefaultEdge> dot =
                        new DOTExporter( new StringNameProvider<>(), null,null);
      FileWriter fw = new FileWriter(new File("java_graph.dot"));
      dot.export(fw, stringGraph);
}

/** Create and simple Graph
 *  @return */
private static UndirectedGraph<String, DefaultEdge> createStringGraph()
{
      UndirectedGraph<String, DefaultEdge> model =
                        new SimpleGraph<String, DefaultEdge>(DefaultEdge.class);
      String top = "top";
      String v1 = "v1";
      String v2 = "v2";
      String v3 = "v3";
      String v4 = "v4";
      String v5 = "v5";
      String bottom = "bottom";
      // add the vertices
      model.addVertex(top);
      model.addVertex(v1);
      model.addVertex(v2);
      model.addVertex(v3);
      model.addVertex(v4);
      model.addVertex(v5);
      model.addVertex(bottom);
      // add edges to create a circuit
      model.addEdge(top, v1);
      model.addEdge(top, v3);
      model.addEdge(top, v4);
      model.addEdge(v1, v2);
      model.addEdge(v2, bottom);
      model.addEdge(v3, v5);
      model.addEdge(v4, v5);
      model.addEdge(v5, bottom);
      return model;
      }
}
```

4. Internal Python DSL : GraphViz libraries

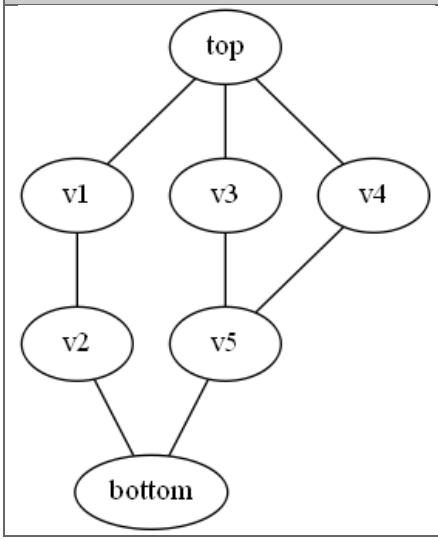
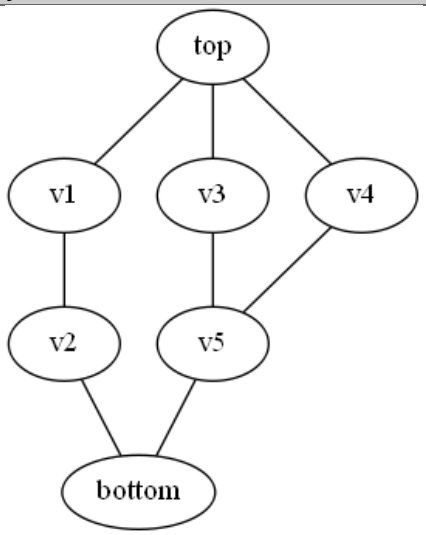
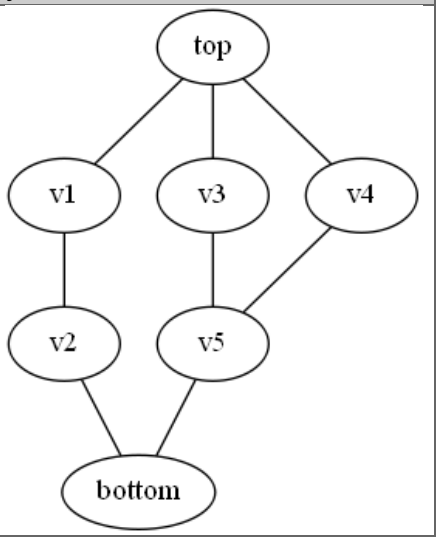
In order to be able to write such a python program, you will first have to install graphviz :
`pip install graphviz`

Then to execute it and write the output into a file, you can execute the follow command :
`python generate_dot.py >> graph.dot`

```
import graphviz as gv

g1 = gv.Graph(format='svg')

""" Nodes """
g1.node('top')
g1.node('v1')
g1.node('v2')
g1.node('v3')
g1.node('v4')
g1.node('v5')
g1.node('bottom')

""" Edges """
g1.edge('top', 'v1')
g1.edge('top', 'v3')
g1.edge('top', 'v4')
g1.edge('v1', 'v2')
g1.edge('v2', 'bottom')
g1.edge('v3', 'v5')
g1.edge('v4', 'v5')
g1.edge('v5', 'bottom')

print(g1.source)
```

generate_dot.py

5. Summary Table

External DSL	JAVA Internal DSL, Dot output	PYTHON internal DSL, Dot output
graph model { **top;** **v1;** **v2;** **v3;** **v4;** **v5;** **bottom;** **top -- v1 -- v2;** **top -- v3;** **top -- v4;** **v2 -- bottom;** **v3 -- v5;** **v4 -- v5;** **v5 -- bottom;** **}**	**graph G {** **top;** **v1;** **v2;** **v3;** **v4;** **v5;** **bottom;** **top -- v1;** **top -- v3;** **top -- v4;** **v1 -- v2;** **v2 -- bottom;** **v3 -- v5;** **v4 -- v5;** **v5 -- bottom;** **}**	**graph {** **top** **v1** **v2** **v3** **v4** **v5** **bottom** **top -- v1** **top -- v3** **top -- v4** **v1 -- v2** **v2 -- bottom** **v3 -- v5** **v4 -- v5** **v5 -- bottom** **}**
		

The three different **dot** files were opened in **GraphViz**.
As you can see, the three graphs generated are exactly the same.

6. Source

http://jgrapht.org **(to download java libraries)**
https://github.com/jgrapht/jgrapht
https://en.wikipedia.org/wiki/DOT_%28graph_description_language%29