Graph Topology Example on LSGi  
Author: Tere Gonzalez, April 6 2017

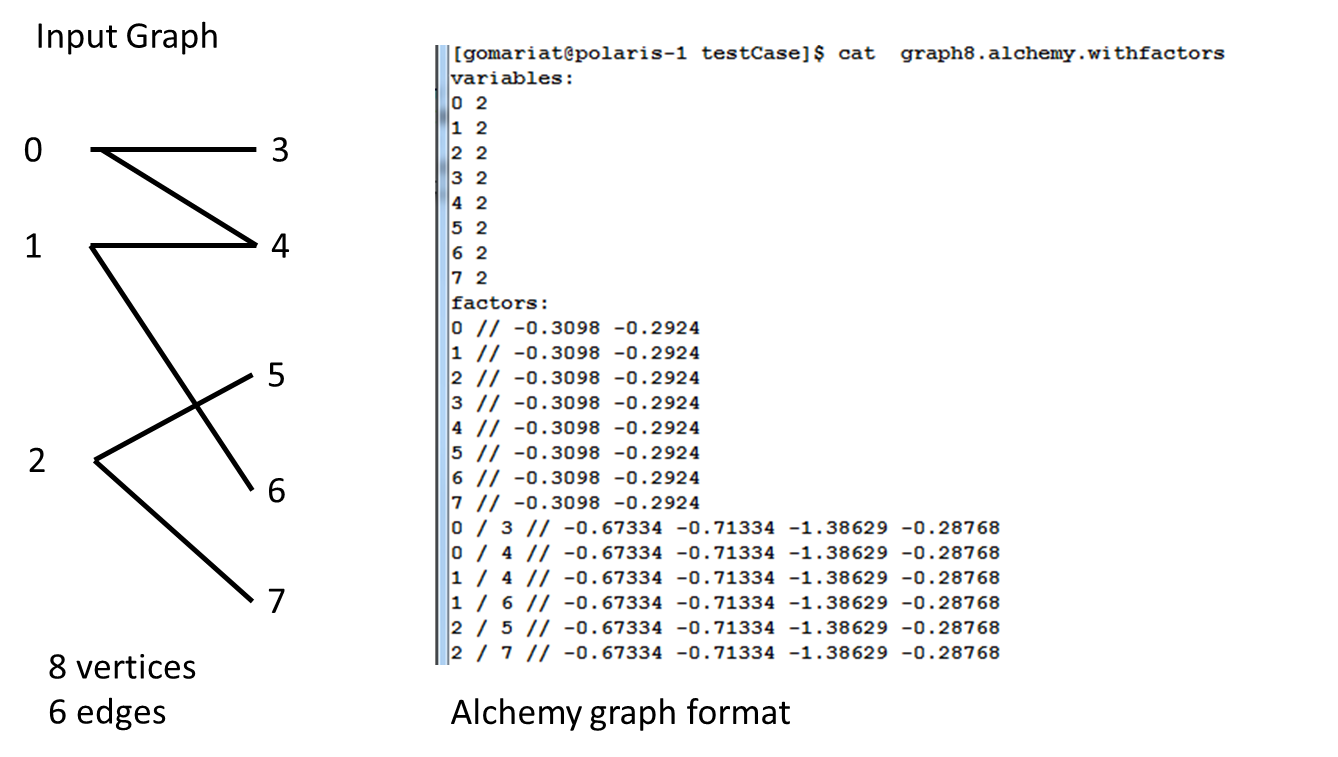
Overview:

This document describes how the input graph is used to build the graph topology index in LSGi that ensures sequential reads for each vertex neighborhood minimizing catches misses.

# Input Graph

The input graph consists on edge list (connected vertices in the graph) plus the edge and unary factors (weights obtained by evaluating factor functions which related the connection between vertices for each vertex state).

Input Graph: 8 vertices, 6 edges with 4 factors.



Note:

1. Format:
   1. List of vertices with the number of variables per vertex.
   2. List of unary factors per vertex, 2 per vertex, 1 for variable 0 and 1 for variable 1
   3. Edge list with edge factors (2 ^ number of variables = 4)
2. Vertices are binary variables for the inference problem., it can take two “states={0,1}”,
3. Unary and edge factor are in log space.

# Binary Input Graph

The previous graph is converted in binary format to speedup the reading of the file as follows:

1.Header:

#number of Vertices (long)

#number of variables or states per vertex (long)

#number of unary factors in bytes (long)

#number of edge factors in bytes (long

2. Content of the file:

for each edge(pair of connected vertices)

fromVertex\_ID (long)

toVertex\_ID (long)

unary\_factor1 (float) state 0

unary\_factor2 (float) state 1

edge\_fractor1 (float) states: (0, 0)

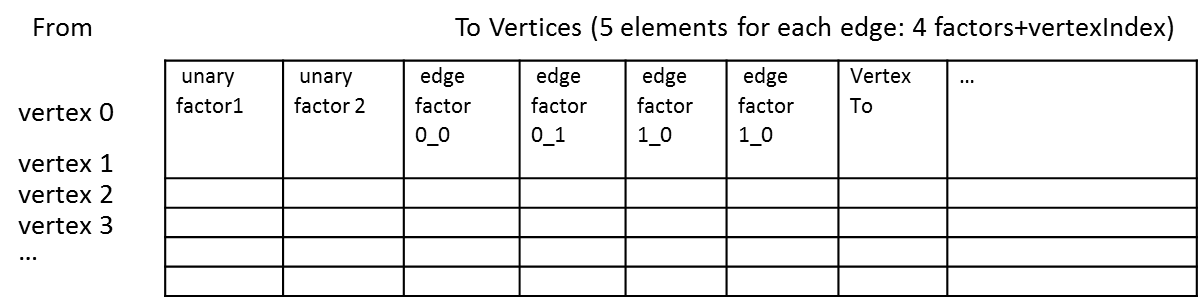
edge\_fractor2 (float) states: (0, 1)

edge\_fractor2 (float) states: (1, 0)

edge\_fractor2 (float) states: (1, 1)

# Topology Index

LSGi builds a graph topology index in each node enumerating all the vertices that will be computed on the node. The index is a materialized view of the edges for each vertex a long with the edge factors. This table ensures sequential read every time that the inference should be computed for each vertex.



The topology index is a big sequential array that layout the list of edges sorted by vertex id. The array contains the unary and the edge factors for each pair of connected vertices. Then, the size of elements of the topology is

Unary factors = #number of vertices \*2

Edge factors = #edges \* 5

Total size = (unary factors +edge factors)\*2; we multiply by 2 because we save both directions n the index, for example from v1 to v3 and v3 to v1.

Then, to compute the number of edges is:

(Total size of the topology – (#numbeOfVertices \* 2))/10 = total edges