Architecture and Design of The Java Kekulé Cell Application

Aaron Germuth and Alex Aravind

Department of Computer Science
University of Northern British Columbia
E-mail: (germuth,csalex)@unbc.ca

July 8, 2013

Abstract

This report outlines the implementation details of our Kekule Cell Application. You should be familiar with Kekule Theory [1, 2] to fully understand this document.

1 The Bit Vector

A Bit Vector (or Bit array, bitmap, bitset, bitstring, etc) is an array data structure that compactly stores bits. It can be used to implement a simple set data structure. Although the Bit Vector is an array, the entire set can be contained with a single binary integer. Every 1 indicates the presence of something, and 0, the absence. For example, 0001 0101 (or 21) is a set of three elements. Node 1, node 3, and node 5.

Because the entire array is within an integer, it is very easy and computationally quick to perform certain operations. You can add or remove a node from your set by simple addition / subtraction. You can iterate through the set by starting at 1 and performing left bit shifts. You can use bitwise AND to take the intersection of two sets, or to check whether a set contains a given element (Figure 1). You can add two sets together with bitwise OR. You can even take the symmetric difference between two sets with XOR.

Whenever you can distinguish elements of your set with a single number, and your sets are not large enough to overflow an integer, you should consider using bit vectors. They are extremely compact, versatile, and quick.

In this application, bit vectors represent sets of nodes, as in the example above. They are also used to represent edges (a set of nodes with 2 elements) of graphs.

2 The Cell

A port assignment is the set of ports which have a double bond in a particular Kekulé structure. In this application, cells have two purposes.

1. A cell can hold an array of port assignments, one for every possible resonance structure of that molecule. It can be used to fully describe the electrical switching behaviour
2. A cell can hold an array of edges. Each edge is represented using a Bit Vector. In this way, a cell con constitute an entire graph.

3 Main

Stolen from Methodology Document:

We have grouped many solutions for obtaining graphs from a cell together. The procedure is as follows. First, Hesselink's approach is performed. Instead of simply returning the first graph which satisfies the cell, we keep searching until we have tried all possibilities within the vertex limit. We then parse through our list of applicable graphs. First all graphs are checked for their highest degree. Graphs with a degree more than 3 on normal vertices or more than 2 on ports are removed.

Next, graphs which are disjoint are fixed. This procedure is outlined in section 4.2. The graph is parsed for detection of 3 cycles (cycles of length 3). The steric hinderance of a 3 cycle is too unstable to be present within a real molecule. For this reason, we extend such cycles to 5 cycles, which are commonly seen in aromatic hydrocarbons.

We then add to the long list of trimmed graphs based on all the template molecules. If one section has not found an applicable graph at this point, we attempt to add edges. We take every graph from every classification below the missing one, and try to add possible edges to the graph. Only edges which will exceed the maximum degree of the graph are considered. In the case of adding an edge between the ports, the ports can often be 'extended' first to prevent the degree from exceeding our limit. From this, we display the graph with the lowest amount of nodes to the user. (currently in text form).

4 Genetic Algorithm

A genetic algorithm is used to search for graphs which apply to a given cell. Resulting graphs are subject to restrictions in the Methodology document[1] The algorithm begins by randomly generating a population of 500 graphs. Each starting graph has anywhere from rank + 1 to 16 nodes, and 4 to 20 edges. This makes up the first generation and is inserted into the algorithm.

[1] Interval vertices are restricted to a degree of <4 , ports are restricted to a degree of <3 , cycles within the graph must be of size >4 , all graphs must be connected, and no intersection of edges.

A population is a group of graphs, ordered by their fitness. In each iteration, the survivors of last generation's population must be chosen. This is done by selecting top 90 graphs (fitness-wise), and 10 random graphs. Random graphs are added to ensure genetic diversity in the population. This sub-population of 100 graphs now undergoes genetic operations such as mutation and crossover. We must have 500 by the end of each iteration in order to have the same size population that we started with. To acheive this, 200 new graphs are added from mutation and 200 from crossover.

4.1 Mutation

Mutation involves randomly perturbing graphs to obtain new graphs. Since we are generating 200 mutants from 100 graphs, it is likely each graph will be mutated twice. Mutations include:

1. 20 % chance to add a new node
2. 20 % [2] chance to remove a random node, and all edges to it
3. 40 % chance to add a random edge [3]
4. 40 % chance to remove a random edge

4.2 Crossover

Crossover begins by randomly selecting 2 out of the 100 graphs and reproduction. These graphs are combined to create a new child graph. Again, since 200 new child graphs are created, it is likely each graph will be a parent twice. Children receive all edges with both parents share, and have a 50

4.3 Fitness Function

Fitness values are integers given to a graph. The fitness is calculated by iterating over the graphs cell compared to the supplied input cell. For every port assignment they share, the fitness is incremented by one. For every port assignment that is missing, or is extra, fitness is decremented one point. If a cell is empty, with no port assignments [4], it is given a fitness value of -10, in attempt to discourage that graph from evolving onwards.

[2] The option to remove a node can only take place if no new node was added to this mutant. By this logic, the actual percentage of removing a node is .80 * 0.20 or 16 %. I need to decide whether this is what I want...

[3] Edges are randomly chosen until an edge is found that can be added without breaking our restrictions on the degree of vertices.

[4] This can happen when the genetic algorithm creates a graph which has a (non-port) vertex with no edges to it (degree 0). There is no way for this graph to have a Kekulé structure since the lone vertex can never have a double bond.

References

[1] W. H. Hesselink, Graph Theory for alternating hydrocarbons with attached ports. Indagationes Mathematicae, Elsevier, 24:115141, 2013.

[2] W.H. Hesselink, J.C. Hummelen, H.T. Jonkman, H.G. Reker, G.R. Renardel de Lavalette, M.H. van der Veen, Kekule Cells for Molecular Computation. Cornell University Library Online, 2013.

[3] M.H. van der Veen. π-Logic. PhD thesis, University of Groningen, May 2006.

[4] A.J. Heeger, S. Kivelson, J.R. Schrieffer, and W.-P Su. Solitons in conducting polymers, Rev. Mod. Phys.,60:781, 1988.