Practical Assignment 3

Available Since: May 13, 2011    Due Date: June 1, 2011, 2:30 p.m.
You are permitted and encouraged to work in groups of two.

Topic of this assignment is the layout of series parallel graphs and the detection of biconnected components.

General Hints

• implement all assignments in the framework from the course website

• create one class for each assignment in your own package:
gd.assignments.lastname1lastname2
    on change of groups create new packages according to group memberships
    your class should implement the interface gd.material.GDLayouter
    unless mentioned otherwise
    ensure getName() returns the names of the algorithm and group members

• to integrate your class into the GUI, add your class to the list in
gd.material.MainWindow.layouts

• rescale the coordinates that your algorithm produces

• for the construction of example inputs, you can either handcraft example graphs in visone, or use one of the random graph generators (File→create)

• send your results and questions to uwe.nagel@uni-konstanz.de
Exercise 1: Bertolazzi-Layout  
15 Points

Implement the layout algorithm of Bertolazzi for series parallel graphs.

- Ensure that in your implementation the decomposition tree is determined explicitly i.e. there should be one function that produces a decomposition tree from a given graph.
- Use the construction of the decomposition tree to determine whether the given graph is series parallel. If it is not, express that in the `canLayout()` function.
- Explain in the comments to your implementation why its asymptotic run time is linear in the number of nodes.

Exercise 2: Biconnected components  
5 Points

Implement the detection algorithm for biconnected components.

- Highlight affiliations to the individual components by different node colors.

To visualize your results, you can use the command

```java
new OrganicLayouter().doLayout(graph);` to layout the graph.
```

For changing the node color, you can use e.g. the following command:

```java
((Graph2D) graph).getRealizer(n).setFillColor(Color.red);
```