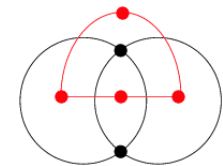


## 2.2 Embedding the Diagram

The final piece of the jigsaw:

### **Converting from a dual graph to an Euler diagram**

1. Triangulating the dual
2. Routing contours around the triangulated dual
3. Improving layout



This section will be illustrated with the general diagram embedding software

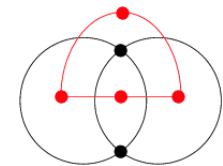
Go to the tutorial web page:

<http://www.eulerdiagrams.com/tutorial/AutomatedDiagramDrawing.html>

At the top of the page there is a link to the user guide and download page for the tool.

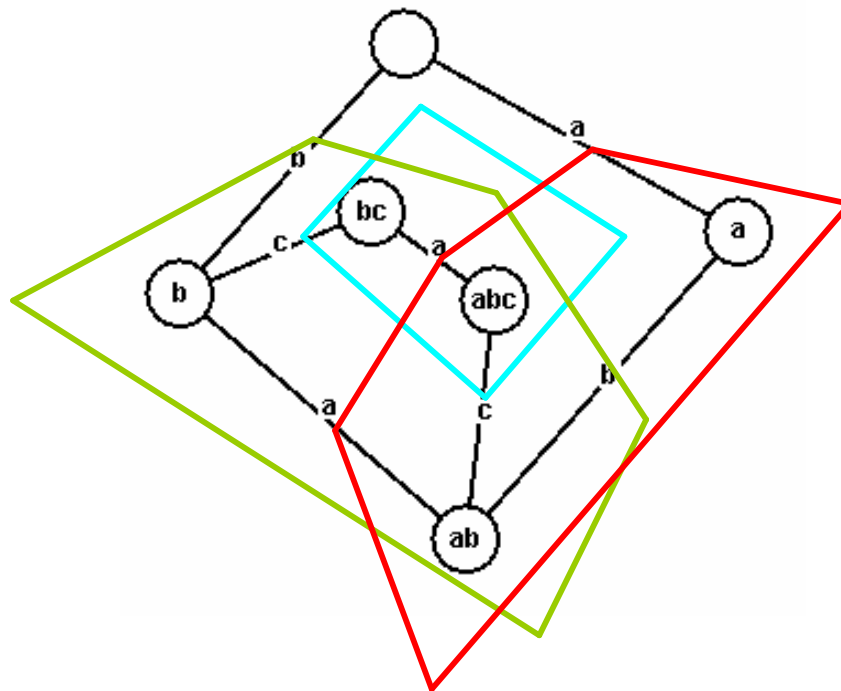
Abstract descriptions can be entered (press 'F1') to generate superduals, the processes described in Section 1.3 and this section can then be performed using 'F2' to 'F5' (see the 'Euler Diagrams' menu).

The contours can be seen by pressing 'c'.

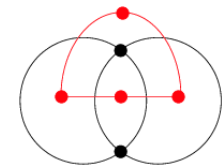


## Motivating the Use of a Triangulation

Contours must pass through faces, but the faces may not be convex. If straight line, or arbitrary routing is used, the result can be unwanted extra minimal regions. Below we have split zones, but zones not in the abstract description can also be created

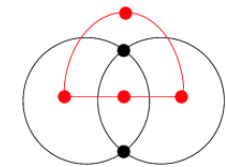
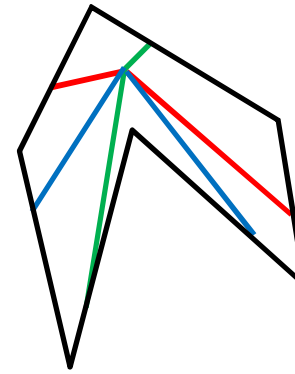
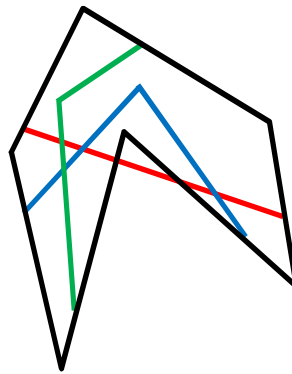


**Generating  
Abstract Description**  
 $\emptyset$  a b ab bc abc



## Requirements of Routing Through a Face

1. All lines in a face must meet at the same point
2. Lines must cross (or meet) once and only once
3. Lines must not leave the face

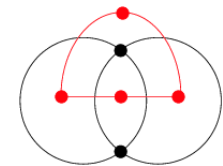


## How Triangulation Solves These issues

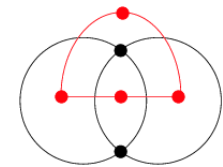
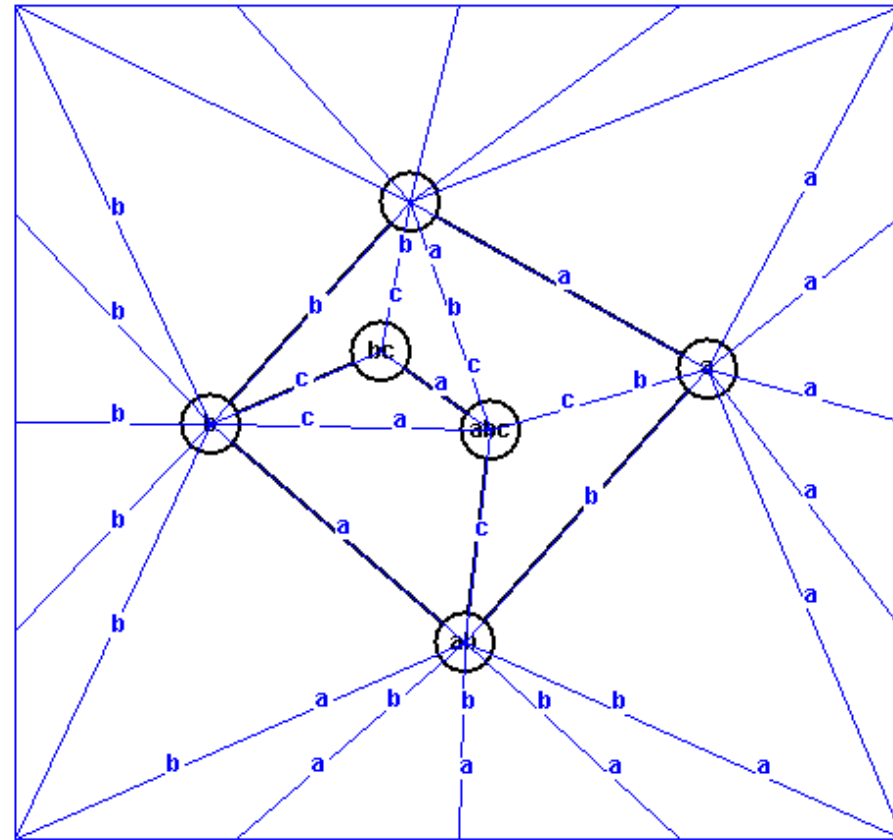
Lines pass through triangles in a straight line.

1. Pick a meeting triangle
2. The meeting triangle is only triangle where edges cross
3. Find a path through triangles from the edge to the meeting triangle

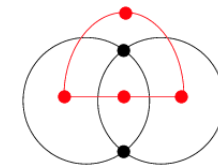
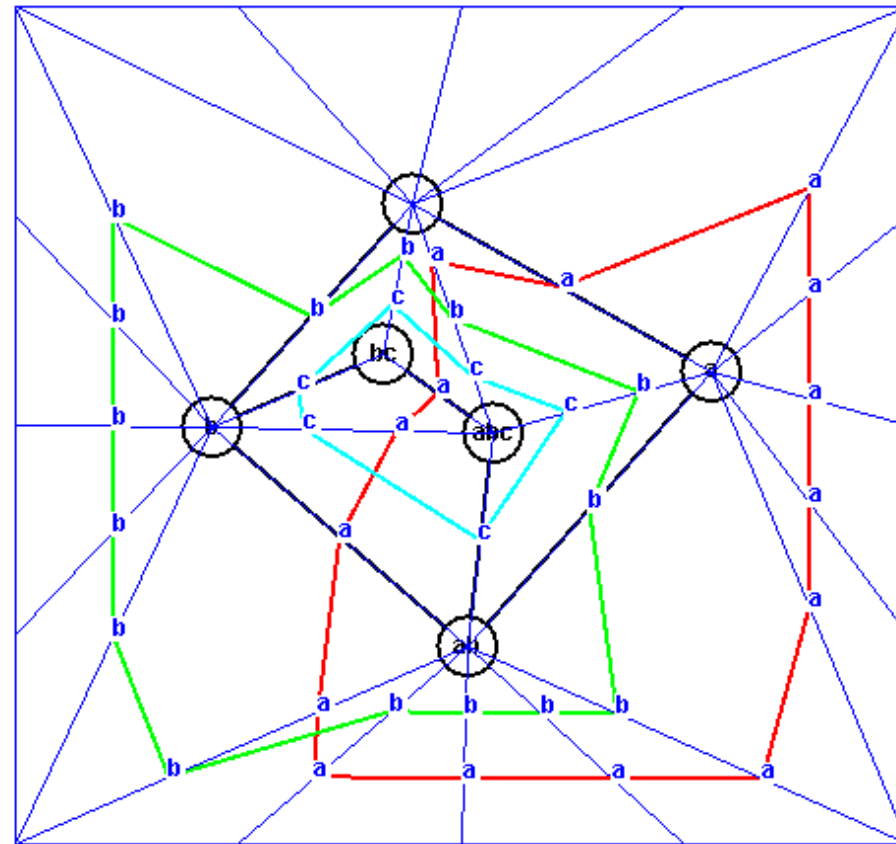
The lines on the triangle edges must be ordered to avoid crossings in all but the meeting triangle



# Example triangulation



# Example with the contours



## Improving the layout of the triangulation

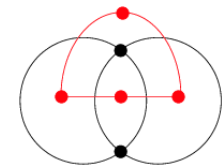
Long, thin triangles produce jagged, hard to comprehend Euler diagrams

Hence we apply a modified Spring embedder to get the internal angles of the triangles more even

An extra force is applied to ensure that planar graph is maintained, by repulsing nodes from edges, ensuring that they will not cross

**Try this with the General Embedding software:** create a dual (or use the default dual), press 't' and the triangulation can be seen, 'F5' then applies the modified Spring Embedder

Task





## Improving the layout of the contours

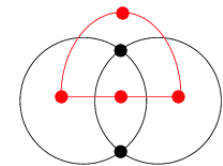
The point where contours route through points in the triangulation edges is flexible, as long as order is maintained

As a simple way of smoothing curves, the angles at these points can be made even. Hence, a search for the best points along the triangle edge for each curve can be made

We know the ideal angle for a point ( $180^\circ/\text{number of points on a curve}$ ), so at each stage this is a local test, making it relatively fast

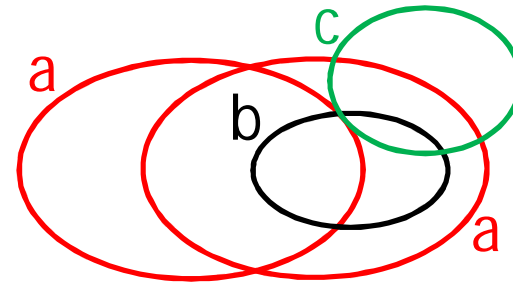
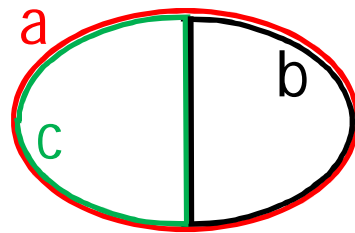
### Task

**Try this with the General Embedding software**, create a dual, press 'c' and the curves can be seen, 'o' then applies this optimization



## Some problems to test your understanding of the layout of Euler diagrams

What abstract description do the below diagrams have?



Embed the abstract description " $\emptyset$  a b c ab ac"  
breaking three different well-formedness conditions

Try the seven standard examples from Session 2.1 in  
the General Embedding tool. Experiment with  
adding and removing edges

